

RECADA / Airholder — полное описание проекта (v59)

Идея · архитектура · файлы и форматы · рабочий процесс · конфигуратор · прототипирование · испытания ISO 11154 · аксессуары · наём · жизненный цикл · ISO 9001 · инструмент реверса · редактор изделия · данные и обратная связь · инфраструктура · материалы · скорость · удалённая работа · права · монетизация · Airholder Ready · датасеты · риски · дорожная карта — ревизия 59 · 11 сентября 2026 · единый документ со сквозной нумерацией

Содержание

№	Раздел
—	Легенда: термины и сокращения
—	История ревизий
1	Идея и исходная ситуация
2	Архитектурный принцип и слои
3	Файлы и форматы: как устроена истина
4	Как проектирует человек
5	Конфигуратор (Unity + Shopify) и инструкции
6	Прототипирование: четыре ступени
7	Аксессуары: вторая ось
8	Кадровая лестница: рост по результатам в логах
9	Наём в разных странах: рынки и ставки
10	Жизненный цикл: две оси
11	Изменение деталей: где дорого
12	ISO 9001
13	Физические испытания и ISO 11154
14	Открытая платформа сообщества
15	Роль FreeCAD и инженера: кто что делает
16	Инструмент реверса recada.py — построен и проверен
17	Редактор изделия: багажник как параметр, а не как геометрия
18	Библиотека деталей и сервер: система с обратной связью
19	Инструкция по монтажу как артефакт изделия
20	Два мира: B-Rep и сетка
21	Данные: источники, каталоги, единая структура
22	Конкуренты как источник данных
23	Проверки уровня изделия: размещение, доступность, высота, настройка
24	Дизайн продукта: что даёт система
25	Обратная связь из поля: рекламации, опыт покупателей, соцсети
26	Модель версий: уроки Onshape
27	Инфраструктура: решения по железу
28	Граф зависимостей конструкции и автоматизация
29	Финансовая модель — каркас
30	Управленческий слой: роли подробно

№	Раздел
31	Материалы и технологии
32	Внешние ресурсы: что берём и чего не берём
33	Скорость: что ещё ускоряет вывод багажника
34	Удалённые столы: способы, выбор, география
35	Найм: три профиля и тесты самой системой
36	Права на результаты и коммерческая тайна
37	Монетизация: модели А, Б, В
38	Программа «Airholder Ready»
39	Поток вместо кодинга: измерить и уменьшить итерации
40	Управление проектом: доска по воротам вместо диаграммы Ганта
41	Датасеты CAD и CAD-агенты: что есть в мире и что из этого наше
42	Гибка листового металла в датасетах: BenDFM и KUL-bend
43	Риски и узкие места
44	Дорожная карта

Легенда: термины и сокращения

Короткий словарь для чтения документа. Каждый термин объяснён одной строкой — подробности в соответствующих разделах.

Термин	Что это простыми словами
JSON	Текстовый формат данных «поле: значение» — язык параметров системы (params.json, паспорта); правится блокнотом
YAML	«JSON для людей» — тот же смысл, но читается как список с отступами; у нас — язык инструкций по монтажу
spec.yaml / SDD	Спека задачи до геометрии: зачем деталь, с чем стыкуется, проверяемые условия приёмки, чего не делаем; CI строит из неё приёмочный прогон; основа сдельной оплаты (Spec-Driven Development)
Python / build123d	Язык программирования / библиотека, на которой написаны генераторы — код, строящий геометрию из параметров
Генератор (generator.py)	Скрипт, читающий params.json и паспорт и строящий деталь; читается как техпроцесс
Паспорт (vehicle/accessory passport)	JSON с точками крепления, углами и поверхностями машины или аксессуара — всё, что нужно знать о чужом объекте
Скелет / MasterParams	Каркас из точек и осей с главными параметрами (высота, углы) — мост между машиной и изделием
Git	Система контроля версий: полная история «кто, что, когда и зачем изменил»; основа всей работы
Gitea	Наш собственный веб-сервер для Git (аналог GitHub, но на нашем железе): репозитории, права, ревью, канбан
Репозиторий (repo)	Папка проекта под управлением Git со всей историей
Коммит (commit)	Одно сохранённое изменение с обязательной причиной («зачем») в сообщении
Дифф (diff)	Наглядная разница между двумя версиями файла: что добавлено, что убрано, строка за строкой; «читаемый дифф» = изменение видно глазами (height: 60 → 80), поэтому истина хранится текстом
Ветка (branch)	Параллельная линия работы для экспериментов; main — главная, защищённая
PR (Pull Request)	Запрос «принять мои изменения в main» — точка ревью и подписи; в main нельзя писать напрямую
Tag (git-tag) / Релиз	Именованная метка версии; релиз детали = подписанный тег, только из него идёт производство
LFS (Large File Storage)	Расширение Git для тяжёлых файлов: в истории — лёгкий указатель, тело — в хранилище рядом; для сканов, STEP, отчётов
zippey	Фильтр, позволяющий Git читать содержимое .FCStd (FreeCAD) и показывать осмысленные диффы
CI (Continuous Integration)	Серверный конвейер: по коммиту/тегу сам собирает геометрию, гоняет проверки и выкладывает артефакты
Артефакт	Файл, построенный CI из истины (STEP, GLB, PDF, BOM): не редактируется — пересобирается
Валидатор / DFM	Автопроверка «производимо ли»: правила гибки Blechcon, зазоры, упаковка, крепёж; красный = стоп
B-Rep / STEP-топология	Способ хранения точной геометрии: тело → оболочки → грани → рёбра → вершины; из него и извлекаются размеры при обмере
AAG (граф смежности граней)	Граф «грань → соседняя грань»: отсечение торцевых граней расщепляет деталь на верхнюю и нижнюю оболочки — основа автоматического распознавания гибов
Скелет детали / Medial Axis	Срединное представление без толщины: пластины + линиигиба + толщина одним параметром; результат обмера и основа параметризации
FTG (граф переходов гибки)	Свёртка AAG: гиб = ребро графа, фланцы = узлы; объединяет соосные гибы, разорванные разгрузками, в один технологический переход
BA / BD (припуск и вычет)	Формулы длины развёртки через K-фактор и нейтральную ось (ANSI/DIN) — считаются аналитически, без CAD

Термин	Что это простыми словами
Kerf (ширина реза)	Металл, съедаемый лазером/плазмой; закладывается в раскрой, иначе деталь выходит меньше номинала
Разпрузка (Entlastungsschnitt / relief)	Вырез у краягиба, чтобы металл не рвался при гибке; в системе вычисляется правилом по t/R , а не копируется из оригинала
Пружинение (springback)	Металл после гибки частично распрямляется; у нержавеющей стали заметно — угол на прессе задают с запасом
FEM / CalculiX / Gmsh	Расчёт прочности и собственных частот (модальный); Gmsh строит сетку, CalculiX считает — бесплатные
CFD / OpenFOAM	Расчёт обтекания воздухом (аэродинамика); тяжёлый, применяется разово вне конвейера
STEP / DXF	Обменные форматы точной геометрии: STEP — 3D для Blechcon и от OEM, DXF — плоские развёртки и сечения
STL / GLB	Сетки из треугольников без точных размеров: STL — печать прототипов, GLB — показ в конфигураторе
.FCStd	Файл FreeCAD; у нас двойной статус: скелеты руками = истина, просмотрные сборки от CI = артефакт
BOM (Bill of Materials)	Спецификация: из чего состоит изделие, массы, цены — генерируется CI
catalog.json	Контракт между CI и конфигуратором: список деталей со статусами, ссылками на GLB и правилами
SKU / Rev (ревизия)	Артикул детали / версия конструкции внутри артикула (Rev A, B...)
FFF (Form-Fit-Function)	Правило: посадка и функция не изменились → новая ревизия того же SKU; изменились → новый SKU
VDI / Kasm	FreeCAD в браузере: программа работает на нашем сервере, человеку летят только пиксели; файлы не покидают сервер
Контейнер / Docker	Изолированная «коробка» с одним приложением; VDI-сессия = одноразовый контейнер
WireGuard / VPN	Шифрованный туннель в нашу сеть; ключ = допуск, отзыв ключа = мгновенное отключение человека
2FA / TOTP	Второй фактор входа: код из приложения на телефоне
Webhook	Автоматический сигнал одной системы другой («остаток = 0») — триггер для скриптов CI
LLM / Claude / MCP	Языковая модель как помощник; MCP — протокол, через который Claude управляет FreeCAD и инструментами
Промпт	Задание модели обычным языком («поднять платформу на 40 мм»); меняет параметры, не геометрию напрямую
GDPR / GeschGehG / StVZO / ISO 11154	Регуляторика: защита данных ЕС / закон ФРГ о коммерческой тайне / правила допуска ТС / стандарт испытаний багажников
CM1 / CM2	Маржа изделия / маржа с учётом стоимости разработки — уровни финансовой модели
CPQ	Configure-Price-Quote: инструмент дилера «собрал конфигурацию → цена → КП → заказ в производство»
OCCT / OpenCASCADE	открытое геометрическое ядро (B-Rep, булевы, скругления, STEP); то же ядро внутри FreeCAD; у нас вызывается напрямую
ОСР	Python-обёртка OCCT из пакета cadquery-ocr; на ней написан recada.py; версия ядра 7.9.3
recada.py	единый файл инструмента реверса: обмер, оболочки, кинематика, растяжение, операции, приёмка; CLI и браузерный интерфейс
headless	работа без графического экрана — на сервере, в CI, пакетно
Streamlit	библиотека браузерного интерфейса на Python; рабочее окно инструмента
three.js	библиотека 3D-графики в браузере; наш просмотрщик сборок
HLR (Hidden Line Removal)	модуль OCCT: проекции тела с видимыми и скрытыми линиями — штриховые виды для чертежей и инструкций, вектор

Термин	Что это простыми словами
XCAF	слой OCCT для структуры сборки: имена деталей, дерево, положения, цвета в STEP
BREP (формат)	внутренний формат ядра; версия меняется с релизом; внутри процесса — да, на границах — только STEP
Три критерия приёмки	метрология (точность), целостность (свободные рёбра = 0, солид ≥ 1), распознавание порталом поставщика; далее: посадка по скану, входной контроль, полевой
Метрология (point-sampling)	расстояние от точек эталона до поверхности кандидата, двусторонняя выборка, порог 99.9 % в 0.05 мм
Свободное ребро	ребро с одной гранью — дыра в оболочке; их ноль = деталь замкнута
Виртуальный щуп	точка — нормаль $\times t$ попадает на поверхность $\rightarrow$ грань оболочечная; метод расщепления листа на две стороны
Оригами-кинематика	жёсткие пластины на шарнирах по осям гибов; поворот угла — поворот поддевера
К-фактор	положение нейтрального слоя в гире (0.33-0.5); задаёт припуск BA; калибруется по ответам портала
Портал Blechcon	веб-портал поставщика резки и гибки: принимает STEP, считает развёртку; третий критерий приёмки
product.json / project.json	истина изделия и проекта: параметры и журнал операций; геометрия — артефакт из них
Aufbaurichtlinien	руководства производителей для строителей надстроек (VW, Ford BEMM, Stellantis): нагрузки, зоны, запреты, крепёж — первый источник ограничений
OEM-CAD / JT / CATIA	заводская геометрия кузова; форматы JT и CATIA конвертируются в STEP один раз (Inventor открывает JT)
FBX	формат визуальных моделей (Blender, Unity); машина как сетка с условным масштабом
ICP / Open3D / trimesh	регистрация скана в систему координат (ICP), библиотеки работы с сетками
WebRTC / стриминг выюпорта	передача кадров в браузер; «наружу только пиксели» — геометрия не покидает сервер
WASM / OCP.wasm	ядро в браузере: OCCT и OCP, скомпилированные в WebAssembly; инструмент у клиента без сервера
Zoo / KCL	zoo.dev — CAD с кодом как истиной (язык KCL), облачный движок в США; ближайший по философии проект
Nutenstein / T-паз	сухарь в паз алюминиевого профиля; интерфейс крепления аксессуаров и кронштейнов
CAPA	Corrective and Preventive Action — цикл корректирующих действий ISO 9001; у нас: рекламация $\rightarrow$ правило валидатора
Струхаль (St)	безразмерное число частоты срыва вихрей; расчёт гула на поперечинах формулой без CFD
DIN 33402 / ISO 2768	антропометрия (доступность) / общие допуски на чертежах
EX44 / Storage Box / restic	сервер Hetzner (i5-13500, 64 ГБ, FSN1) / хранилище для бэкапов (Хельсинки) / программа резервного копирования
Микроверсия	неизменяемая запись изменения со ссылкой на родителя (Onshape); у нас — операция журнала с хешем родителя
Определение / результат перестроения	что хранится (список операций) / что кэшируется и всегда перестраивается (B-Rep, сетка); истина — определение
Устойчивое изменение	операция адресует элемент по постоянному признаку, а не по позиции — применима к другой версии исходника
Переигрывание журнала	применение всех операций к новой ревизии исходной детали; Ranger-кронштейн v2 получает те же правки
Геометрический отпечаток	адресация грани или отверстия по типу, нормали, центру, площади — замена постоянных ID, которых в OCCT нет

Термин	Что это простыми словами
Проблема топологического именования	номера граней меняются при перестроении; известная беда FreeCAD; у нас решается отпечатком для листовых деталей
Ветка / слияние журнала	варианты изделия как ветки списка операций; слияние без конфликтов — «источник побеждает» (правило Onshape)
Геометрический diff	сравнение двух версий: грани добавлены, удалены, сдвинуты — подсветкой в просмотрщике
FeatureScript	язык собственных операций Onshape; подтверждение, что «операция как код с параметрами» — верная абстракция

История ревизий

Рев.	Дата	Что добавлено
v25	31.08.2026	цеховая практика, сборочные проверки, закрыты классы «физических» и «геймдев» идей
v26	03.09.2026	15а инструмент реверса recada.py; 15б редактор изделия; три критерия приёмки; три исправленные ошибки
v27	03.09.2026	15 переписан: роль FreeCAD и инженера, идея workbench; 15в библиотека и сервер; риски и дорожная карта по практике; Zoo
v28	03.09.2026	как Zoo стримит; свой движок: ядро нет, обвязка да
v29	03.09.2026	ядро напрямую без FreeCAD; создано с ИИ, проверено реальностью; просмотрщики на ОССТ
v30	03.09.2026	решение: каскадный вьювер на сервере, наружу только пиксели; достаточность EX44; защита данных
v31	03.09.2026	версии ядра 7.8/7.9/8.0; STEP на границах; STEP с именами как требование
v32	03.09.2026	15г инструкция по монтажу как артефакт: разбор Defender 110, инструменты, четыре слоя
v33	03.09.2026	15д два мира: B-Rep и сетка; FBX, GLB из ОССТ, Blender
v34	03.09.2026	15е данные: шесть источников, два каталога, единая структура, заводская система координат, четвёртый критерий
v35	03.09.2026	опция: реверс из STL; входной контроль как пятый критерий
v36	04.09.2026	15ж конкуренты как источник данных; история ревизий
v37	04.09.2026	15з проверки уровня изделия: размещение, доступность, высота, настройка
v38	04.09.2026	15и дизайн продукта: что даёт система — покупатель, конструктор, эксплуатация, бренд
v39	04.09.2026	15к обратная связь из поля: рекламы, опыт покупателей, соцсети; шестой критерий
v40	04.09.2026	Aufbaurichtlinien как первый источник ограничений; ссылки в правилах валидатора
v41	04.09.2026	легенда дополнена терминами v26–v40 (26 понятий)
v42	04.09.2026	15л модель версий: уроки Onshape — микроверсии, устойчивые изменения, переигрывание, ветки без конфликтов; легенда
v43	06.09.2026	материалы и технологии; учёт времени по журналу

Рев.	Дата	Что добавлено
v44	06.09.2026	внешние ресурсы: OCCT 8.0.1, YouCanNotUnfold, китайские и российские ядра
v45	07.09.2026	скорость: девять идей; удалённые столы; найм: три профиля; права и коммерческая тайна
v46	07.09.2026	монетизация: экосистема аксессуаров; проектирование в системе на данных о машинах
v47	07.09.2026	модель В: производители аксессуаров и самоделкины; классы ответственности
v48	07.09.2026	модель В: дилеры и установщики — B2B у прилавка
v49	07.09.2026	программа «Airholder Ready»
v50	09.09.2026	поток вместо кодирования: методика на Ranger; четыре механизма против итераций
v51	10.09.2026	управление проектом: доска по воротам вместо Ганта
v52	10.09.2026	датасеты CAD и агенты: обзор, лицензии, MM-CAD; свой датасет
v53	10.09.2026	гибка в датасетах: BenDFM, KUL-bend; исправление
v54	10.09.2026	пересборка в единый документ со сквозной нумерацией 1–44; ссылки обновлены
v55	10.09.2026	раздел 42 подробно: BenDFM по файлам, CADFS по ключевым словам, ABC щупом, пружинение, TRUMPF, ESICUP; сводный план
v56	10.09.2026	раздел 42: инструмент прессов и таблицы гибки — WILA/Roller в оригами-кинематику, базы Bendguide/LVD через Blechcon, экспорт из CAM, шесть шагов
v57	10.09.2026	раздел 42: NUMISHEET 2025 с картой 1.4301, бенчмарки пружинения, FEM-корпуса (не наше — зафиксировано), first-part-right, полный реестр внешних источников
v58	10.09.2026	раздел 42: SMCAD, HeteroMF, DOI для скачивания, ESAFORM, трубы (не наше), четыре кластера; приложение «Источники» — 63 ссылки
v59	11.09.2026	раздел 18: как учиться система — три механизма, правило и случай, внешние ресурсы как полигон, портал Blechcon как оракул цены, что знать об оборудовании

1. Идея и исходная ситуация

RECADA (Аугсбург) производит автомобильные багажники Airholder: платформа из алюминиевых экструдированных профилей (профили на складе, сечения нарисованы один раз) + гнутые ноги из нержавеющей стали 1.4301 (2–3 мм); резка и гибка — аутсорс (Blechcon.de). Узкое место — ноги под каждую машину и аксессуары. Цель: система, где на входе крыша (STL-скан с точками крепления или STEP от OEM), на выходе — производственные файлы, обновлённый конфигуратор и инструкции; дизайн управляется по промпту; работа командная с версионированием; знания накапливаются; исполнители — удалённые инженеры из разных стран, отбираемые по результатам тестовых работ.

2. Архитектурный принцип и слои

Источник истины — параметры и код, а не геометрия. Файлы FreeCAD/STEP — артефакты сборки. Это даёт промпт-управление (LLM меняет числа в JSON, не двигает грани), читаемые Git-диффы, командную работу без конфликтов бинарников, автоматическую пересборку производных.

Слой	Содержание
1. Паспорт автомобиля	JSON из STL/STEP (один раз): точки крепления с нормальными, углы, межосевые; для деталей, повторяющих крышу (windshield), — опорные поверхности как сплайн-аппроксимация (не сырая сетка); поле точности geometry_source. Сырой скан — в архив, роль: контрольный эталон для проверки зазоров.
2. Скелет + MasterParams	Точки, оси, плоскости, главные параметры (H, alpha). Всё «твёрдое» читает размеры отсюда. Скелет — мост между паспортом машины и паспортом изделия. Железное правило скелетного метода: детали никогда не ссылаются друг на друга — только на скелет (иначе связи рвутся непредсказуемо); направление зависимостей строго сверху вниз: MasterParams → Skeleton → детали.
3. Каталог профилей	Сечения один раз (импорт DXF); параметр — длина резки; «ширина от длины» — правило.
4. Генераторы (build123d)	Код строит детали от скелета; материал, R3, покрытие зафиксированы; библиотека «кубиков» (bent_flange, hole_array, bolt_joint) прячет сложность.
5. Валидатор DFM+	Правила Blehcon (отгибы, радиусы, развёртка, коллизии) + зазор до крыши ≥ 20 мм (статический запас вместо расчёта деформации кузова) + упаковка по лимитам канала доставки (текущий канал GLS: ≤ 40 кг/коробка — главный лимит, масса считается по коробкам комплекта с раскладкой в BOM; длина до ~2 м практически, пороги надбавок — из тарифа GLS в rules/) + крепёж (таблица болт/гайка, проверка зацепления). Красный — артефакты не собираются. Физическая калибровка правил гибки (FEA-in-the-loop): CalculiX симулирует сам процессгиба и показывает, при каком радиусе напряжение на внешнем волокне грозит микротрещинами у 1.4301; расчёт делается один раз для сетки «толщина × радиус × марка» → таблица допустимых R в dfm_blehcon.json → валидатор смотрит в неё мгновенно. Так справочное « $R \geq 1 \times t$ » заменяется на подтверждённое расчётом и практикой поставщика. Проверка против реальной оснастки цеха (CAM-in-the-Loop в практической форме): у Blehcon запрашивается список доступных пуансонов и матриц (радиусы, ширина V-паза, максимальная длинагиба) и заводится таблицей в dfm_blehcon.json — валидатор отвечает «R2.5 нет, ближайший R3, пересчитываю K-фактор» и проверяет минимальную длину фланца, чтобы деталь не проваливалась в матрицу, ещё до отправки заказа. Генерация G-кода для их пресса не входит в систему — станок программирует поставщик.
6. CI-сборка	STEP/DXF → Blehcon (перед заказом автоматический раскрой: SheetNest / nest2d / CAD-N, для больших партий u-nesting на Rust с Python-интерфейсом — отходы, проверка счёта, учёт ширины реза kerf из dfm_blehcon.json); GLB → конфигуратор; студийные рендеры (Blender headless) → Shopify; PDF/HTML → инструкции; BOM → цены и склад.

recada-cad/ | — vehicles/ паспорта машин | — products/ конфигурации (params.json) | — catalog/ профили, крепёж | — accessories/ аксессуары и адаптеры | — rules/ DFM, правила | — ci/ пайплайны сборки | — generators/ build123d + кубики Версионирование: Gitea (self-hosted) + Git LFS; .FCStd — через zippey (читаемые диффы). Релиз = git-тег с персональной подписью ревьюера. Новая машина = паспорт + params.json — механизм масштабирования.

3. Файлы и форматы: как устроена истина

Никакого собственного формата: истина детали = два обычных файла. `params.json` — числа и слова (SKU, машина, материал, высота, отверстия), правится блокнотом/промптом. `generator.py` — Python/build123d, читается как техпроцесс: «лист 120×340×3 → скруглить углы → отверстия по точкам паспорта → фланец 90° R3 → нижний гиб под угол крыши из паспорта». Числа приходят из трёх источников (`params` / паспорт / каталог материалов), в коде их почти нет. FCStd — не источник, а результат (как `.exe` при исходниках): CI собирает его для просмотра. Исключение: скелеты и эталоны реверса, построенные руками, — полноценные FCStd-исходники в Git.

Дерево построения не исчезло — оно переехало в текст: порядок строк = порядок операций, но вместо хрупких ссылок на номера граней (Topological Naming Problem) — устойчивые правила (`edge="top"`, точки по именам) и порождающие конструкции (циклы по точкам паспорта, условия). Плюс история с «почему»: каждый коммит несёт причину/промпт («поднять платформу 40 мм — бокс задевает антенну»), чего нет ни в одном GUI-дереве; это же — QMS-след и обучающие данные системы.

Сборки: тот же принцип уровнем выше

Сборочный JSON описывает состав и привязки, но не координаты: детали публикуют именованные интерфейсы («sole», «head» с плоскостью, отверстиями, нормалью), сборка говорит «посадить sole на mount_front_L». Вместо решателя констрейнов (Inventor) — детерминированное вычисление позиций из интерфейсов: ничего не «желтеет» при изменениях. Крепёж — свойство соединения: `bolt_joint()` сам ставит болт по оси, считает длину из пакета, кормит BOM. Промпт «поднять платформу на 40 мм» меняет один параметр скелета — всё пересобирается. Иерархия: деталь → сборка изделия → конфигурация клиента, один формат и конвейер. Цена подхода: нет свободной кинематики (шарниры — позой-параметром или в песочнице).

Роли форматов и «труба»

JSON+Python (истина) → STEP/FCStd (точная геометрия) → DXF (развёртка) → STL/GLB (сетки)
Вниз — всегда, автоматически. Вверх — только реверс-инжиниринг (дорого, один раз, на входе).

STEP — «PDF мира CAD»: точный, но мёртвый; выход к Blechcon и вход от OEM. STL/GLB — сетки без размеров: печать и показ, никогда для производства и измерений. У каждого производного файла есть родитель-истина (SKU+ревизия в имени); файл-сирота — нарушение процесса. При конфликте прав всегда верхний по трубе, нижние пересобираются. Разнородные исходники (скан машины + чужой STEP багажника) никогда не стыкуются напрямую — каждый переводится в паспорт-JSON, стыковка на языке паспортов; ноги — материализованный мост между двумя паспортами.

FFF-правило (Form-Fit-Function)

Посадка и функция не изменились → новая ревизия того же артикула (взаимозаменяемо, склад смешивает). Изменились → новый SKU с политикой замены. Защищает от зоопарка артикулов и от тихой несовместимости. Исполняется машиной: валидатор сравнивает именованные интерфейсы между ревизиями; «функция сохранена?» — единственный пункт, остающийся подписи человека.

Справочник типов файлов: что лежит в Git и как с этим работать

Тип файла	Что это	В Git?	Кто и как правит
<code>params.json</code>	Параметры детали/изделия: SKU, машина, материал, размеры, статусы; рядом — <code>spec.yaml</code> : намерение, интерфейсы, ограничения приёмки, non-goals (контракт задачи)	Да, обычный текст — главный источник истины	Инженер (редактор/веб-форма), LLM по промпту; каждое изменение — коммит с причиной; в <code>main</code> только через PR

Тип файла	Что это	В Git?	Кто и как правит
generator.py и recada_lib	Python/build123d: код построения геометрии; библиотека кубиков (bent_flange, hole_array, bolt_joint)	Да, обычный текст — источник истины	Senior/владелец (+LLM с ревью); пробы в ветке convert/*, squash-merge при приёме
Паспорта: vehicle_*.json, accessory_*.json	Точки крепления с нормальями, углы, опорные поверхности (сплайны), интерфейсы, масса/ЦТ, geometry_source с точностью	Да — источник истины	Создаются один раз при оцифровке (человек или ИИ-агент, подпись человека обязательна); уточняются при пилотных установках
Сборочные JSON	Состав и привязки по именованным интерфейсам (без координат)	Да — источник истины	Инженер/LLM; те же PR-правила
rules/: DFM, материалы, тарифы	Правила Blechcon, materials.json (1.4301, R3), лимиты GLS, таблица крепежа, пороги FEM	Да — источник истины	Владелец/lead; пополняются из рекламаций и ревью технолога Blechcon (цикл CAPA)
YAML инструкций	Шаги монтажа: деталь, крепёж, момент затяжки	Да	Инженер; CI рендерит PDF/HTML/Unity-сцены сам
.FCStd (FreeCAD)	Два разных случая! (а) Скелеты и эталоны реверса — построены руками; (б) просмотрные сборки — построены CI	(а) Да, через zipреу (читаемые диффы) + LFS; (б) НЕТ — артефакт	(а) Владелец/senior в FreeCAD; (б) не редактируется никогда — пересобирается
STEP / DXF	Точная «мёртвая» геометрия: обмен с Blechcon и приём от OEM/исполнителей	Входящие — архив (LFS) как сырьё; исходящие — НЕТ, артефакты CI с SKU+ревизией в имени	Не редактируются; исходящие уходят Blechcon только из CI по тегу; входящие — через извлечение в паспорт
STL / GLB	Сетки без размеров: печать прототипов (STL с ID получателя), конфигуратор (GLB огрублённый)	НЕТ — артефакты CI	Не редактируются; для производства и измерений непригодны принципиально
Сканы крыш (STL), STEP от OEM	Сырьё оцифровки	Архив в LFS (один раз)	Не редактируются; роль — контрольный эталон зазоров и источник паспорта
ВОМ, catalog.json, рендеры, PDF-инструкции	Производные для склада, конфигулятора, клиентов	НЕТ — артефакты CI (хранилище артефактов; релизные — вечно)	Не редактируются; изменения — только через правку истины и пересборку
Отчёты испытаний и FEM	Протоколы физических испытаний, эталонные расчёты (статика, модальный)	Да (LFS) — прикрепляются к релизному тегу	Только добавляются; связь «ревизия ↔ протокол» = QMS-след

Три железных правила для всей команды: (1) редактируется только истина (JSON, .py, паспорта, rules, YAML, скелеты) — всё остальное пересобирается, файл-артефакт руками не трогается никогда, даже «быстро поправить»; (2) у файла без SKU и ревизии в имени нет права покидать сервер; (3) при расхождении версий прав всегда тот файл, что выше по трубе истины — нижние пересобираются, а не чинятся. Нарушение правила 1 (ручной STEP в производство мимо rapams) — главный риск смерти системы, поэтому отправка на Blechcon технически возможна только из CI-артефактов релизного тега.

4. Как проектирует человек

Принцип «шаблон, не копия»: дерево/генератор строится из конструктивного замысла (база → полки → гибы → крепёж, каждый размер осмыслен и привязан к параметру), а не как копия всех артефактов конкретной штампованной детали — копия хрупка и ломается при больших изменениях, шаблон держит. Путь 1 — новый тип детали: творческий поиск руками в песочнице FreeCAD (мышкой, как привык) → готовая конструкция становится измеримым эталоном (объём, масса, точки поверхности) → перевод в код по шагам (контур → сверка → гибы → сверка → отверстия → сверка), каждая попытка автоматически сверяется с эталоном; пробы живут в отдельной ветке и сплющиваются в один коммит при приёмке; 3 неудачи LLM подряд → эскалация человеку. Путь 2 — повседневный: правка чисел в params (редактор/форма/промпт) → пересборка → просмотр; CAD превращается в место просмотра. Путь 3 — LLM-переводчик: человек описывает словами/эскизом, Claude пишет build123d по образцам библиотеки, CI собирает, метрология возвращает расхождение, LLM читит.

Классы деталей по стоимости перевода

Класс	Перевод в систему	Трудозатраты
Профили (алюминий, прокат)	Импорт DXF-сечения + extrude; параметр — длина	час на каталог
Гнутый лист (ноги, адаптеры)	Реверс техпроцесса в код; быстрее с библиотекой кубиков	часы-дни
Литые/скульптурные	НЕ переводить: паспорт интерфейсов (20 мин) + STEP как мёртвый референс; офсеты от опорных поверхностей при необходимости; полный реверс — только если планируется семейство размеров	минуты
Покупное (крепёж)	Библиотеки стандартных деталей build123d	0

Правило границы внутри детали: форма — руками (один раз), интерфейсы и адаптация — кодом. Люди в системе остаются для четырёх ролей, где машина слаба: постановка задачи (знание реального мира), топология («как вообще решить»), суждение («валидно, но глупо» — подпись ревьюера), обучение системы (опыт → правила). Час «обвести контур» система обнуляет; час «понять, как обойти рейлинг» становится вечным активом.

Вход открыт для любого CAD: исполнитель рисует в SolidWorks/Inventor/Catia/FreeCAD — сдаёт STEP + чертёж с размерами; перевод в код не зависит от исходного CAD. Требовать build123d от конкурсантов = сузить пул вдесятеро. Лицензии: проверить чужую лицензию невозможно — декларация в договоре переносит ответственность, а VDI-сессия снимает проблему полностью (легальное рабочее место без единой лицензии у исполнителя) и очищает цепочку поставок для ISO/GeschGehG. Субподряд безопасен по построению: оплата за деталь, прошедшую валидатор; аккаунт персонален (сессии из двух стран видны в логах); на лестнице доверия субподрядчик не растёт.

Инженерный задел: что уже проверено практикой

Реверс-инжиниринг с метрологической приёмкой (point-sampling по контрольным точкам): Bracket_L — 100% совпадение; Bracket_R — зеркало, 99.12%; wind_side_v2 — 100.000% по 88 точкам; Part55 → новый Bracket_88_W30 с полным DFM-контролем; узел Defender 110 (LRDE-2BR1) — 99.9994%, SheetMetal-дерево от скелета. Материал везде 1.4301 2B, t=3.0, K-фактор из таблицы. Критический урок сессий: установка networkx включила SheetMetal Unfolder V2, который показал истинные углы гыба 99.389° и 135° вместо ошибочно измеренных V1 80.6° и 45° (дополнительные углы) — ошибка с прямыми производственными последствиями; правило: развёртки только через V2. Словарь конструкции зафиксирован: полка (Flansch) — боковые грани с отверстиями, стенка (Steg) — центральная грань с логотипом.

Выученные технические правила (в skills): SMBendWall падает на прерванных гыбах → Revolution или реконструкция из face-призм; extend задавать до навешивания дочерних стенок (иначе TNP рвёт привязки); упавший конструктор оставляет null-объект — удалять

немедленно; boolean cut ненадёжен для верификации — только point-sampling через distToShape; перед поиском кромок — полный touch()+recompute (кэш SheetMetal не сбрасывается сам); привязки только через SubShapeBinder (защита от TNP); для деталей с формовками — двухконфигурационное дерево (плоская модель для развёртки + финальная с формовкой). Модули к установке: SheetMetal 0.8.22 (Solid-to-SheetMetal, folding marks для прессы Blechcon, индикация битых ссылок), Defeaturing WB (срезать фаски в зонегиба → развернуть → вернуть вырезы), Curves WB (чистка B-Spline-граней из STEP Inventor), networkx. Материалы — Material Definition Sheet (таблица R/T → K-фактор вместо жёсткого K=0.42): одна таблица питает и Claude, и параметрические модели. Unattended Unfold — headless-развёртка для CI. Точка входа репозитория — CLAUDE.md: описание изделия, материалы по умолчанию, какие skills когда применять.

Валидатор с автопочинкой: следующая ступень DFM

Правила оцифровываются в три слоя: числа → dfm_blechcon.json (R=1xt, мин. полка 15 мм для нержавеющей, отступ отверстия отгиба, Entlastungsnut, зазоры $\geq t$, лимиты листа, GLS-габариты); исправления → features.py (генераторы разгрузок: овал, Entlastungsnut, угловая разгрузка); методики → skills (markdown: «как думать»). Валидатор не только говорит «ошибка: отверстие близко к гибу», но предлагает исправление: «конфликт → предлагаю relief_obround, вот превью» — в RECADA-workbench это красная подсветка + кнопка «исправить», в CI тот же код headless блокирует выпуск неисправленного. Калибровка библиотеки — по реальному производству: принятые Blechcon детали с их правками разгрузок обмеряются (простой реверс), формулы «размер разгрузки от t/R/d» зашиваются в features.py; каждое письмо-исправление от Blechcon — строка в dfm_blechcon.json, и ошибка не повторяется ни у кого. Дизайн по правилам перестаёт зависеть от того, помнит ли инженер PDF поставщика.

Реверс без GUI: STEP → Python-код (метод и стек)

Дерево построения как таковое не нужно — нужен Python-код: он хранит ту же последовательность операций, но без хрупких ссылок на грани (TNP), с циклами и условиями, диффами и историей «зачем». Просмотровое дерево CI может собрать для глаз, но истина — код. Вся труба реверса работает headless (build123d/OCP — то же ядро OpenCASCADE, что у FreeCAD): без GUI, без MCP-сессий, повторяемо и пригодно для CI (десять STEP за ночь).

Алгоритм автоматического обмера (step_probe.py): обход B-Rep через TopExp_Explorer → классификация граней (плоские = полки/стенки, цилиндрические = гибы, узкие торцевые = толщина) → построение AAG (атрибутированный граф смежности граней, networkx): узлы — грани, рёбра — сопряжения → отсечение торцевых граней расщепляет граф на две изолированные эквидистантные оболочки (верх/низ), их сопоставление даёт нейтральную поверхность и толщину t → извлечение параметров гибов (радиус, угол, ось; numpy: нормали, PCA осей), отверстий (цилиндры с осями и D), внешнего контура. Выход — measurements.json.

Скелет детали — главный результат обмера и основа параметризации

Ключевое решение: обмерщик выдаёт не «список размеров», а СКЕЛЕТ детали — срединное представление без толщины: пластины (плоскости фланцев с контурами) + линиигиба (оси, углы, радиусы) + толщина как ОДИН параметр ($t = 2 \times$ расстояние от срединной поверхности до границы). Реверс превращается из «восстановления формы» в «восстановление скелета», и это даёт три эффекта. (1) Толщина перестаёт быть геометрией: смена 3 мм → 2 мм — правка одного числа, а не перестроение всех поверхностей (актуально: применяются обе толщины). (2) Скелет детали и общий Skeleton сборки говорят на одном языке — плоскости и линии, поэтому связка «изменил N платформы → пересчитались углы фланцев ноги» становится геометрически прямой: линиягиба лежит на пересечении плоскостей, заданных скелетом сборки. (3) Параметризация осмысливается почти сама: длины пластин, углы между ними и положения линийгиба — ровно те величины, что меняются от машины к машине, а артефакты производства (кern-метки, случайные скругления) в скелет не попадают вовсе — схлопываются вместе с толщиной. Medial Axis фильтрует шум за инженера.

Реализация — без «настоящего» Medial Axis Transform (в OCC штатной функции нет, устойчивое построение срединных поверхностей на реальных B-Rep нетривиально): скелет строится из уже существующего AAG — пары эквидистантных граней дают срединные плоскости, цилиндры гибов дают срединные оси и радиусы. Результат идентичен, а код — тот, что и так спроектирован. Скелет детали сохраняется в `measurements.json` как основная структура; дальше на него навешиваются параметры (`spec.yaml`) и правила (`rules/`).

Кривизна как устойчивый детектор гибов и формовок

Второй, независимый от описания поверхности критерий: для каждой грани считаются главные кривизны и производные величины — плоскость даёт гауссову $K=0$ и среднюю $H=0$; цилиндрический гибо — $K=0$ при $H \neq 0$, причём направление нулевой главной кривизны и есть осьгиба, а вторая кривизна даёт $1/R$; формовки и выштамповки — $K \neq 0$ (неразвёртываемые участки). Практическая ценность прямая: критерий работает независимо от того, как STEP описал грань — при экспорте из Inventor плоскости и цилиндры часто приходят B-spline-аппроксимациями, где формального «цилиндра» нет и поиск по типу поверхности проваливается, а кривизна есть у любой поверхности. Бонус: $K \neq 0$ сразу помечает деталь как содержащую формовку, то есть требующую двухконфигурационного подхода, — ещё до попытки развернуть. Порядок в обмерщике: сначала опрос типа поверхности через OCC, при неудаче — замер кривизны.

Автопроверка корректности развёртки (Гаусс-Бонне): для правильной плоской развёртки сумма угловых дефектов по контурам даёт 2π ; отклонение означает топологически неверную развёртку — складки или самонакладки. Одна формула в приёмке развёртки: дешёвая страховка, ловящая грубые провалы анфолдера автоматически, без человеческого взгляда на DXF (метрологию не заменяет).

Что автоматика НЕ делает: выбор, какое число — параметр (высота, меняется от машины), какое — правило ($R3$ всегда), какое — формула (ширина от длины), а какое — артефакт производства, который не воспроизводится. Это конструктивный замысел; его расставляет человек или Claude со skills, читая `measurements.json` как текст. Далее — `spec.yaml` + `generator.py` + `params.json`.

Детектор разгрузочных вырезов (Entlastungsschnitte / bend reliefs)

Разгрузки предотвращают разрыв и смятие металла при гибке, но усложняют топологию — именно они вызывают «прерванные гибы», на которых падал SMBendWall и путался Unfolder V1. Распознавание в три шага, встраивается в тот же обмерщик: (1) дешёвая эвристика по контурам и рёбрам гранигиба — канонический гибо имеет 1 контур и 4 ребра; больше одного контура = сквозной вырез внутригиба, больше четырёх рёбер = вырезы по краям; (2) паттерн в AAG — идеальный гибо смежен ровно с двумя плоскими фланцами, разгрузка добавляет ему в соседи мелкие торцевые грани или разрывает цилиндр на несколько соосных (поиск подграфа в уже построенном графе, ничего нового не требуется); (3) UV-проекция границгиба на цилиндр — отклонение от прямоугольника даёт точные размеры выреза (глубина, ширина, отступ); применяется только к найденным случаям.

Ключевое отличие от «просто копирования»: в системе разгрузка — это ПРАВИЛО, а не форма из оригинала. Генератор вычисляет её по толщине, радиусу и типу узла (`features.py`), а не переносит контур. Поэтому у детектора две роли: сигнал «этот узел технологически требует разгрузки» (иначе валидатор выдаст конфликт и предложит `relief_obround`) и источник калибровки — обмер разгрузок на деталях, принятых Blehcon, даёт реальную зависимость «размер от $t/R/d$ », которая зашивается в `features.py` навсегда. Третий выигрыш практический: знание «на этом гибепрерывание из-за разгрузки» позволяет автоматически выбрать проверенный обход развёртки (Revolution вместо SMBendWall либо Defeaturing: срезать вырезы → развернуть чистый гибо → вернуть вырезы плоскими) — то, что сегодня решается руками в каждой сессии, становится ветвью в коде. Инструменты те же: OCP (TopExp_Explorer, BRepAdaptor_Surface) + networkx; отдельный pythonocc-core не нужен — то же ядро уже доступно через OCP.

Гибрид «точные числа + зрение» и агентная петля коррекции

Ключевое дополнение к обмерщику: геометрия и семантика извлекаются разными средствами и соединяются. OCC/AAG/FTG дают точные числа (толщина, радиусы, углы, координаты отверстий) — в этом машина безупречна. Рендер детали в 4 ракурсах (проекции + изометрия из build123d) плюс мультимодальная модель дают семантику ролей: «это разгрузка у основаниягиба», «это фальц для жёсткости края», «это П-образный кронштейн», «это артефакт штампа, воспроизводить не нужно». Именно семантики не хватало: AAG измеряет, но не знает, что параметр, что правило, а что артефакт. Работает сегодня без обучения моделей и без датасетов — рендер, взгляд, разметка measurements.json, генерация.

Агентная петля автокоррекции (штатный режим реверса): экстрактор → LLM пишет generator.py → CI собирает STEP → сверка с исходным → расхождения возвращаются модели числами («вырез смещён на 2 мм, строка 45») → правка → повтор до сходимости. Стоп-правило прежнее: 3 неудачные итерации подряд → эскалация человеку с дневником попыток (иначе модель крутится вечно). Судья — только числовая метрология point-sampling $\geq 99.9\%$ и точные интерфейсы; визуальный дифф (наложение рендеров оригинала и сборки с подсветкой расхождений) используется как навигация по ошибке и объяснение человеку, но НЕ как критерий приёмки: деталь может «выглядеть одинаково» при расхождении в полмиллиметра на посадочном месте.

Эволюционный подбор параметров — для тех 10–20% случаев, где распознавание сдаётся (свободные радиусы, скульптурные переходы): шаблон пишет человек, числа подбирает оптимизатор (scipy / CMA-ES, без ML), минимизируя ту же метрологическую невязку point-sampling. Дёшево, детерминированно, встраивается в CI как ночной прогон.

Оригами-кинематика и модальная сигнатура

Кинематическая развёртка («жёсткое оригами»): деталь представляется как жёсткие плоские пластины (фланцы) и одномерные шарниры по осям гибов; последовательные повороты вокруг осей до нулевых углов дают плоскую заготовку. Один алгоритм даёт два выхода сразу: траектория решателя (последовательность поворотов) — это и есть порядок гибки, а конечное состояние — заготовка. Метод закрывает сразу две задачи: развёртку без анфолдера (углы шарниров = углы гибов, конечное состояние = контур заготовки) и автоматическую проверку самопересечений при складывании — ту самую Bending Feasibility, которую иначе считает регрессивная развёртка. Реализация — чистая геометрия на numpy (последовательные повороты + тест пересечений); физические движки (PyBullet/MuJoCo) для жёсткого оригами избыточны.

Модальная сигнатура как финальный валидатор реверса: для оригинала и для реверснотой модели считаются первые ~5 собственных частот (CalculiX). Совпадение спектров доказывает механическую эквивалентность, а не только геометрическую: частоты чувствительны к толщине, радиусам в зонахгиба и распределению массы — то, что точечная метрология может проглядеть. Применяется к ответственным деталям (ноги, несущие кронштейны), необязательно для мелочи; инфраструктура FEM всё равно строится под валидатор, так что чек почти бесплатный. Красивое замыкание: тот же модальный анализ, что уже выполнен компанией, служит и приёмкой реверса, и правилом « $f_1 \geq$ порога по Струхалю» против гула на скорости.

Цеховая практика в коде: базы, прокат, допуски, следы инструмента

Базирование от технологических баз (datums), а не от нуля CAD: деталь описывается от тех плоскостей, от которых её меряет ОТК (базы на поверочной плите, щуп КИМ) и от которых её позиционирует оператор у заднего упора листогиба. Тогда переменные генератора называются STOP_DISTANCE_X1, а не offset_x, размеры в коде совпадают с контролируемыми на производстве, и цепочка размеров идёт от базы — меньше накопленной ошибки. В spres.yaml появляется секция datums (базовые плоскости и точки установки).

Направление проката: гиб вдоль волокна даёт микротрещины на внешнем радиусе у нержавеющей стали — при радиусах, близких к минимальным, риск реальный. Правило связывает конструкцию и раскрой: если осьгиба параллельна направлению проката (оно задаётся ориентацией детали на листе при нестинге), валидатор либо требует увеличить минимальный радиус, либо переориентировать деталь в раскройной карте — и записывает предупреждение в комментарий генератора.

Толщина как распределение, а не число: реальный лист «3 мм» бывает 2.93. Обмерщик измеряет толщину в N точках и выдаёт номинал плюс разброс; в генератор идёт номинал, в спеку — допуск с assert-проверкой, К-фактор считается от фактической толщины. Обмерщик перестаёт спотыкаться на «плавающей» толщине.

Следы матрицы как источник технологии: при воздушной гибке на внешнем радиусе остаются две параллельные линии сопряжения цилиндра и плоскости — расстояние между ними выдаёт ширину V-канала, на котором деталь гнули. По чужой детали восстанавливается не только геометрия, но и технология, а радиус сразу привязывается к стандартному ряду матриц ($V = 6, 8, 10, 12, 16$ мм) — прямое дополнение к таблице оснастки поставщика. Проверка коллизий уточняется: не «согнётся ли вообще», а «согнётся ли на этом прессе в этом порядке» — деталь не должна задевать верхнюю траверсу на 2-м и 3-м гibe (габаритные модели инструмента запрашиваются у Blechcon; фирменные каталоги Amada/Trumpf не нужны). В задел без приоритета: распознавание замков «шип-паз» и зазоров под сварку — актуально, если появятся сварные узлы (сейчас соединения болтовые).

Рассмотрено и отклонено (чтобы не возвращаться по кругу): GNN-распознавание граней (BRepNet/CADNet/Sheet-metalNet) — нет открытых моделей под листовую металл, нужен размеченный корпус в тысячи деталей, точность вероятностная; ваш класс деталей распознаётся правилами детерминированно. ML-предиктор пружинения на FEA-датасетах (ANSYS/LS-DYNA — платные) — решает задачу, которую уже решает гибщик на своём прессе; наш путь — строка в dfm_blechcon.json из практики Blechcon. Специализированная VLM «картинка → код» (CAD-Coder) — research-модели слабее связки «точные числа OCC + рассуждение сильной модели». EvoCAD (популяция вариантов с VLM-жюри) — дорого по токенам и оптимизирует визуальное сходство вместо размерной точности. Gaussian Splatting с телефона — точность на бликующем металле далека от требований к интерфейсам; практичная версия (фотограмметрия оболочки + штангенциркуль по крепежу) уже принята; вернуться при оцифровке чужих аксессуаров «в поле». Analysis Situs, pythonocc-core отдельно, CadQuery — дублируют то, что уже в стеке. Инверсный CSG («воздушный слепок»: вычесть деталь из габаритного параллелепипеда и разбирать отрицательный объём на примитивы) — опирается на булевы операции, которые по нашему же выученному уроку на сложных телах возвращают бессмыслицу; радиусы и отверстия и так берутся прямо с цилиндрических граней. Синтетический лазерный скан STEP + нейросети облака точек (CAD-Recode / Point2CAD) — осознанная потеря точности ради устойчивости к дефектам STEP, которые лечатся healing-ом в две строки OCC. Дифференцируемый CAD (SDF + градиентный спуск) — технически рабочий, но решает ту же задачу, что принятый оптимизатор CMA-ES, требуя дифференцируемого рендера и подбора гиперпараметров; вернуться, если CMA-ES упрётся по скорости. Синтез программ через SMT-решатели (SyGuS/Z3) — «доказанная корректность» недостижима: решатели работают с логикой и арифметикой, а не с B-Rep; наша гарантия — измеренная (метрология + спрес-прогон), а не доказанная. Латентная диффузия CAD-деревьев (GenCAD-3D) и спектр Лапласа-Бельтрами — обучение на миллионах программ / математика для поиска похожих форм; плоскости и цилиндры распознаются напрямую. PINN для динамического К-фактора — уравнения пластичности ради числа, которое берётся из Material Definition Sheet и калибруется по принятым Blechcon деталям. RL-агент для развёртки (обучение подбору последовательности разгибания) — задача уже решена детерминированно дважды: регрессивная развёртка и оригами-кинематика дают порядок гибов и проверку самопересечений за секунды и воспроизводимо. Нейро-символический DSL с AST-валидацией через Z3 — тревога «LLM ошибается синтаксически» снята дешевле: код исполняется в CI (синтаксис падает мгновенно и возвращается модели), семантику ловят валидатор и метрология. INR/SIREN (нейросетевое поле заполненности) — потеря точности ради градиентов там, где точные плоскости лежат прямо в B-Rep. Квантово-вдохновлённый генетический алгоритм — обёртка над обычным эволюционным поиском; принятый CMA-ES делает ту же работу с проверяемой сходимостью. Персистентные гомологии (TDA) — из идеи взята только суть (инвариант целостности) в виде чек-листа сохранности элементов, без алгебраического аппарата. Оптические каустики (трассировка лучей для замера радиуса), голономия/параллельный перенос (двугранный угол получается скалярным произведением нормалей в одну строку), тепловой поток по уравнению Фурье для поиска вырезов (перечисление внутренних контуров дешевле на два порядка) — экзотика ради экзотики. Карта мартенситного превращения ($\gamma \rightarrow \alpha'$ при холодной деформации) — физика верная, но для

расчёта нужна симуляция деформации, для которой уже надо знать, где гибы: круговая логика; реальная ценность темы у нас другая — наклёп в зонахгиба влияет на коррозионную стойкость и магнитность готовой детали, это вопрос качества продукта, а не реверса. Класс «фундаментально-физических» методов рассмотрен целиком и отклонён (фаза Берри, поток Риччи/Ямабе, калибровочная теория дефектов, полиномы узлов, обратная задача рассеяния, голографический принцип, симплектическая механика, топологические краевые состояния, функционал Онсагера, сложность Колмогорова): все они решают три задачи — углы и радиусы гибов, развёртка, поиск вырезов — которые уже закрыты кривизной, оригами-кинематикой и счётом контуров с машинной точностью за секунды. Точность системы сегодня упирается не в математику распознавания, а в физику и данные: К-фактор реального листа, компенсация пружинения на конкретном прессе, допуски исходного STEP, точность скана крыши — их эти методы не уменьшают. Читаемость кода (циклы вместо повторов) достигается инструкцией в промпте и ревью, а не вариационной задачей. Класс «геймдев-математических» подходов также рассмотрен и отклонён: IK-риг совпадает с уже принятой оригами-кинематикой (приятное подтверждение естественности решения); SDF-реймаршинг, поток Риччи, shape grammars и Wave Function Collapse, MCTS-поиск кода, воксельные октодеревья, клеточные автоматы — это ранее отклонённые классы в новой упаковке (дифференцируемый CAD, нейро-символический DSL, потеря точности ради устойчивости); краудсорсинговый Foldit требует аудитории игроков, которой нет. Из теории Морса взято только следствие, и так вытекающее из скелетного подхода: строить деталь в коде от базовой плоскости к периферии, а не в произвольном порядке.

Критерий «100% копия с параметрическими изменениями» (в rules/): реверс завершён, когда выполнены все четыре условия — (1) интерфейсы (отверстия, посадочные плоскости, углы) совпадают точно; (2) метрология point-sampling на номинальных параметрах $\geq 99.9\%$ против исходного STEP; (3) прогон на ИЗМЕНЁННЫХ параметрах зелёный у валидатора и спес-теста (доказательство, что параметрика живая, а не декорация); (4) чек-лист сохранности элементов: отверстий было N — стало N, вырезов и разгрузок было M — стало M (считается напрямую: сквозные отверстия — парные цилиндрические грани, вырезы — внутренние контуры; ловит самую обидную ошибку «забыл дырку»); (5) список осознанных отличий от оригинала записан в спеку (кern-метки и случайные скругления не воспроизводятся; технологический язычок воспроизводится, но как правило, а не как копия). Оставшиеся доли процента — обычно К-фактор, точность сплайнов и допуски самого исходного STEP; гнаться за ними — воспроизводить чужие ошибки экспорта.

Стек (только то, что уже в системе): build123d/OCF — чтение STEP и B-Rep; networkx — AAG; numpy — векторная математика; метрология — distToShape point-sampling; развёртка — SheetMetalUnfolder V2 headless (FreeCAD здесь импортируемая библиотека, не среда), с миграцией на YouCanNotUnfold (быстрый анфолдер на networkx, drop-in замена) или встроенную развёртку build123d по мере зрелости; нестинг — SheetNest / nest2d / CAD-N / u-nesting (NFP-алгоритмы с учётом kerf). Отвергнуто как избыточное: pythonocc-core отдельно (то же ядро уже есть через OCF), CadQuery (дублирует build123d), Analysis Situs (тяжёлая C++-платформа ради того, что закрывают ~100 строк на networkx). Пружинение (springback): нержавейка 1.4301 сильно наклёпывается — измеренный на готовой детали угол уже включает отпружинивание; компенсация — забота гибщика, но её реальные значения от t/R стоит завести строкой в dfm_blechcon.json из практики Blechcon.

Spec-Driven Development: спека задачи как контракт

Система уже устроена по принципам SDD (истина — спецификация, геометрия — артефакт, валидатор — исполняемая спека ограничений, история «зачем» в коммитах). Дополняем тремя приёмами. (1) Спек до геометрии: для каждой новой детали до первой линии в песочнице пишется spec.yaml — намерение (intent), интерфейсы стыковки (из паспортов), проверяемые ограничения приёмки (масса, нагрузка, зазор, целевая себестоимость по BOM) и явные non-goals (чего НЕ делаем). Спек — файл истины в products/ рядом с params.json. (2) Спек → тест → реализация: CI генерирует приёмочный прогон из constraints автоматически ещё до начала работы; красный прогон на пустой детали — норма, работа исполнителя = сделать его зелёным; приёмка превращается из «проверить сданное» в «позеленить спеку», субъективность ревью сжимается до вопроса «валидно и не глупо?». Спек = формальное основание сдельной оплаты — контракт чище любого текстового ТЗ; спор «деталь не та»

превращается в «спека была неполна», и чинится спека, обогащая шаблон. (3) LLM работает от спеки, не от чата: промпт сначала материализуется в `спес.yaml` (Claude сам оформляет), человек подтверждает за секунды, потом генерация — намерение зафиксировано в Git до реализации, «LLM понял не так» ловится на спеке, а не на геометрии; правило зашивается в `rules/` и в системный промпт RECADA-ассистента. Внедрение: этап 2, рядом с валидатором (генератор приёмочного прогона — малый скрипт CI).

Сборка: проверки уровня изделия

Детали собираются в багажник, поэтому у сборки свой слой проверок — отдельно от геометрии деталей. Позиционирование уже описано (именованные интерфейсы вместо констрейнов, всё от скелета); добавляются четыре проверки, все — геометрия и арифметика, без физических движений и симуляции монтажника.

Проверка	Что делает и зачем
Доступ инструмента (assembly feasibility)	Для каждого болтового соединения проверяется свободный цилиндр доступа: прямой канал от оси болта наружу под головку ключа (диаметр, длина хода) и вылет для установки стержня. Ловит классическую ошибку «спроектировано, но не собирается»; критично, потому что монтаж идёт на крыше в неудобной позе. Заодно проверяется, что деталь заводится на место при уже смонтированных соседях.
Размерные цепи (tolerance stack-up)	Платформа стоит на четырёх ногах: допуск гибки ($\pm 0.5-1^\circ$), допуск отверстий лазера, разброс крыши — сумма даёт перекося или несведение отверстий на последней ноге. Система проходит цепочку «крыша → нога → профиль → следующая нога» по худшему случаю и статистически (RSS) и выдаёт минимальный необходимый овал отверстий. Снимает целый класс рекламаций «не встало».
Автопорядок монтажа	Последовательность выводится из графа связей (нога к крыше → профиль к ногам → аксессуар к профилю) с учётом проверки доступа, а не пишется руками. Инструкция генерируется автоматически и всегда актуальна; Unity-конфигуратор получает готовую последовательность для интерактивной 3D-инструкции. YAML остаётся для человеческих подсказок («не затягивать до установки задних ног»).
спес.yaml сборки	У изделия тоже есть контракт приёмки: суммарная масса, допустимая нагрузка, число уникальных SKU (метрика унификации — чем меньше разных деталей, тем дешевле склад), оценка времени монтажа по числу крепежа, совместимость аксессуаров. Приёмка сборки — такой же зелёный прогон, как у детали.

5. Конфигуратор (Unity + Shopify) и инструкции

Конфигуратор разрабатывается на Unity (WebGL для сайта; из того же проекта — нативные мобильные приложения с AR-примеркой через AR Foundation и десктоп для дилерского киоска/выставки). Контракт CI ↔ фронтенд не зависит от движка — catalog.json: SKU, ревизия, статус, GLB, точки стыковки, варианты, правило замены; Unity грузит GLB в рантайме через glTFast, attach-точки читаются как named nodes. Возможность «техпроцесс в конфигураторе»: Unity-сцена визуализирует сборочные шаги из тех же YAML-инструкций — интерактивная 3D-инструкция вместо PDF. Риски Unity-WebGL: тяжёлый билд (15–50 МБ против 2–5 у three.js) — следить за временем первой загрузки на мобильном (~5 с порог); при провале — гибрид: лёгкая витрина-превью + полный Unity-конфигуратор. GLB сжимается (glTFpack+Draco: десятки КБ), ревизия в имени файла против кэша CDN. Клиенты видят только протегированные релизы. «Светофор нагрузки»: суммирование масс аксессуаров против допустимой нагрузки крыши — рост конверсии, все данные уже в каталоге. Инструкции из YAML (шаг, детали, крепёж, момент) рендерятся CI с актуальными изображениями; конфигурация клиента известна — инструкция генерируется под его заказ. Конфигуратор на базе FreeCAD для розницы отвергнут (секунды на пересборку, ферма контейнеров, мобильные): розница = pre-built GLB, мгновенно; VDI-FreeCAD с панелью RECADA = CPQ-инструмент для дилеров/нестандарта, выход — params.json сразу в производство.

6. Прототипирование: четыре ступени

Ступень	Что проверяет	Срок/цена ошибки
1. Цифровой	Коллизии, FEM (CalculiX+Gmsh в валидаторе), AR-примерка на машине (model-viewer, тот же GLB)	минуты / 0 €
2. Печатный	--target=proto → STL, FDM 1:1 (PETG): посадка, отверстия, помехи; ID получателя выдавлен на нелицевой стороне; принтер 300–500 € окупается первой итерацией	часы / ~2 €
3. Металл (DC01)	Blechcon как прототипная мастерская; material_override: DC01 вместо 1.4301 (2–3× дешевле); маркировка PROTO not for sale	дни / десятки €
4. Пилотный комплект	Боевой материал, установка, фотофиксация → правки в паспорт и DFM-правила: цикл обучения замкнут	недели / сотни €

7. Аксессуары: вторая ось

Боксы, маркизы, Starlink, LED-бары, солнечные панели. Ключ — стандартизованный интерфейс платформы (Т-паз, шаг): аксессуар привязан к интерфейсу, не к машине — спроектирован один раз, подходит всем. Паспорт аксессуара: крепление, объём + зоны исключения, масса и ЦТ (валидатор считает суммарную нагрузку конфигурации), совместимость. Для покупных изделий проектируется только адаптер — тот же конвейер. Оцифровка: STEP от производителя (дают дилерам) или скан — фотограмметрия для оболочки ($\pm 1\text{--}2$ мм, глянец матировать) + штангенциркуль для крепежа (± 0.05); поле точности в паспорте, неточность компенсируется конструктивно (овальные отверстия). Печатная примерка ловит ошибки оцифровки за 2 €. Библиотека паспортов популярных аксессуаров — накапливаемый актив.

8. Кадровая лестница: рост по результатам в логах

Степень	Доступ	Переход
Новичок	Тестовые и простые задачи, только VDI, сдельно	вход свободный
Проверенный	Прямые заказы, ставка выше	3+ принятых работы
Senior	Сложные узлы, Git-доступ, проектная оплата	устойчивое качество работ + коммуникация
Lead/reviewer	Первичная приёмка, наставничество, ТЗ; решает bus factor	отдельный отбор по качеству мышления в комментариях (принцип Питера: лучший чертёжник ≠ лучший ревьюер)

Одна удачная работа — шум, паттерн на 4–6 задачах — сигнал. Система логирует: качество вопросов к ТЗ, итерации до зелёного, поведение при проигрыше, экономическое чутьё. Повышение вычисляется, а не назначается — прозрачность правил сильнее премий на рынках СНГ/Азии. Финальное доверие (доступ к ядру) — личное решение.

9. Наём в разных странах: рынки и ставки

Рынок	€/час	Роль
Индия	4-10	Основной пул: глубина, английский, бюро sheet metal (Пуна/Ченнай/Бангалор), UTC+5:30. Пинг 100-160 мс — фильтровать пробной сессией.
Узбекистан/Кыргызстан	5-12	Русский трек: ГОСТ-школа, лояльность; отсюда вырастут постоянные и lead. Пинг 60-90 мс — VDI комфортен.
Пакистан	3-7	Добавочные руки строго сдельно (свет/интернет — риск исполнителя; PayPal нет).
Вьетнам	5-12	Отложить (язык, UTC+7, пинг 180-250); стратегичен при азиатском векторе производства.
Россия	—	Напрямую нет (платежи, комплаенс); легально — релоканты (Армения/Грузия/Казахстан) или ИП в третьей стране; Kasm по HTTPS работает при любых блокировках.

Модель найма: удалённые инженеры-конструкторы на сдельной основе (оплата за деталь, прошедшую валидатор) с ростом до постоянного сотрудничества. Воронка: 100 приглашений → 30 откликов → 15 тестов → 8 сдали → 3-4 прошли → 1-2 на скамейку; цена постоянного ~300-500 €. Каналы: Upwork (активные приглашения на платный тест), hh.uz, GrabCAD, кафедры вузов, рефералы. Оффер-магниты: платный тест; не нужен софт и мощный ПК (браузерная сессия — заодно легальность и доступ пула без железа); автоматическая приёмка по опубликованным правилам; прозрачная лестница; постоянный поток. Ступенчатый отсев: фильтр-вопрос → платный тест с автоприёмкой (эталон Bracket_L) → лично смотрите 3-4 из ста. Тест включает 10-мин пробную VDI-сессию (меряет реальный канал кандидата).

10. Жизненный цикл: две оси

Конструкция (status)	Склад/продажа (sales_status)
Draft / Черновик — в работе, снаружи не видна In Review / На проверке — валидатор зелёный, PR открыт Released / Выпущена — подпись + тег + Rev; единственный производимый статус Superseded / Заменена — вышла новая ревизия (автоматически); хранится вечно Obsolete / Снята — негодна, преемника нет; производство блокирует CI On Hold / Заморожена — остановлена без решения	Not for Sale — прототипы (PROTO), пилоты Active — производится и продаётся Run-out — superseded, остаток >0: продаём, не дозаказываем; webhook Shopify при нуле включает преемника Discontinued — снята; карточка живёт вечно (гарантия, QR-инструкции) Service Only — по обращению как запчасть к проданному

Политика замены A→B выбирается при релизе: run-out / немедленная (дефект) / параллельно (FFF: та же посадка и функция = ревизия, не новый SKU). Ось 1 движет CI по git-событиям; ось 2 — webhook Shopify + два ручных решения.

11. Изменение деталей: где дорого

Создание дешево; изменение дорого пропорционально зависимостям. Три тяжести: в существующих параметрах (бесплатно) / новый параметр (правка генератора — узкое место квалификации) / слом топологии (= новая деталь; правило «больше N строк — новый шаблон» против костылей). Каскад (развёртка, масса и нагрузка конфигураций, GLB, инструкция, BOM) CI пересобирает сам; управленческими остаются: неприкосновенность интерфейса (изменение = «версия 2 платформы»), явная политика замены на складе, запрет обходов мимо params (STEP на Blehcon — только из CI, технически). Плюс: геометрический дифф в PR (подсветка изменений старый/новый STEP), версионирование библиотечных блоков с явной миграцией, шлюз на изменение released (issue с обоснованием + go владельца).

12. ISO 9001

Система = исполняемый QMS: 8.3 (входы=паспорт+ТЗ, выходы=артефакты CI, верификация=валидатор, валидация=прототипы, изменения=PR+ревизии); 8.5.2 прослеживаемость (деталь → тег → автор → ревьюер → заказ); 7.5 документация = Git; 10.2 CAPA (рекламация → issue → правило валидатора навсегда). Фрилансеры = внешние поставщики (8.4): страница критериев допуска/оценки/отстранения (данные собирает Gitea), объём контроля привязан к лестнице (риск-ориентированность), компетентность (7.2) = конкурсная история, ревьюеру — чек-лист и персональная подпись. GDPR: обработка данных фрилансеров (аккаунты, логи) — пункт в договоре, запись в Verzeichnis, срок хранения логов. Сертифицироваться — когда спросит рынок (B2B/OEM): фундамент ISO-ready, формализация 2–3 мес, ~3–6 т.€. Правило: документируется исполняемое или используемое — иначе театр качества.

13. Физические испытания и ISO 11154

Слепая зона, закрытая внешним аудитом: система проектирования не заменяет испытаний готового изделия. Даже если съёмный багажник юридически классифицируется как груз (Ladung, §22a StVZO — проверить у юриста, не принимать на веру) и не требует ABE/Teilegutachten, отсутствие документированных динамических испытаний по ISO 11154 (включая City Crash, ускорения до ~16g) при аварии превращает Produkthaftung в персональный риск руководства и закрывает вход в B2B-дилерские сети. Действия: (1) выяснить текущий статус испытаний продукта; (2) при отсутствии — отдельный испытательный трек: эталонные испытания базовой конструкции в аккредитованной лаборатории (бюджет — тысячи €, это цена входа в B2B); (3) расширить FFF-правило пунктом «изменение, влияющее на несущую способность, требует переиспытания» — валидатор помечает такие изменения автоматически; (4) прослеживаемость «какая ревизия испытана» система уже даёт бесплатно (git-тег + протокол испытаний как артефакт релиза).

Текущий статус и расчётная база

Компанией уже выполнены прочностной (статический) и модальный анализы конструкции; планируется аэродинамический. Разграничение: расчёт (FEM) и физическое испытание по ISO 11154 юридически не взаимозаменяемы — расчёт говорит «должна выдержать», протокол испытания документирует «выдержала»; для Produkthaftung и B2B нужен физический протокол, а расчёты — сильное дополнение, удешевляющее сертификацию (меньше итераций в лаборатории). Модальный анализ: собственные частоты против частот вихревого срыва на 100–130 км/ч (гул, резонанс, усталость) — превращается в правило валидатора «первая собственная частота ноги $\geq$ порога» (порог по формуле Струхала из скорости и размера профиля; CalculiX умеет модальный анализ штатно). Аэродинамика прагматично: ветровые нагрузки как входные силы прочностного расчёта — нормативные формулы без CFD; частоты срыва — Струхаль; полный CFD (OpenFOAM) — разово для формы windshield и гула, вне конвейера. Существующие отчёты — эталоны калибровки CalculiX-пайплайна: прогнать ту же деталь, сверить частоты и напряжения — валидатор рождается проверенным (логика метрологического эталона Bracket_L, но для физики).

14. Открытая платформа сообщества

Печатаемые аксессуары сообщества (держатели, клипсы) делают платформу «липкой» и служат разведкой спроса (популярная модель → задача на металлическую версию).

Публикуется открытая спецификация интерфейса (измерима штангенциркулем; опубликовать первым = стать стандартом; управление — у вас, модель USB); публичный репозиторий + STL; автопроверка посадки тем же конвейером; категория Community в конфигураторе. Защита (Produkthaftung): CC BY-NC-SA + disclaimer (вы хостинг, не производитель), автомодерация категорий (несущее на скорости — на свой риск/не публикуется), витринная граница «металл протестирован / community на свой страх». В правилах — право Airholder на металлическую версию по мотивам (с упоминанием автора). Сообщество = нулевая ступень кадровой лестницы. Веб-визуализация не раскрывает ИС: в браузер уходит огрублённый GLB («фотография часов, не чертёж»); децимация — параметр CI; точная геометрия покидает сервер единственным маршрутом CI → Blechcon. От обмера купленного изделия не защищает ничто — реальная защита: скорость конвейера, паспорта, библиотека решений, валидатор, скамейка, стандарт.

15. Роль FreeCAD и инженера: кто что делает

Позиция уточнена по итогам практики. Инженер из системы не уходит — он остаётся автором формы. Полноценный CAD мы не пишем: это не наш продукт и не наша экспертиза. Но новые детали делать надо, и делать их надо руками — в FreeCAD. Инструмент recada.py не заменяет конструктора, он снимает с него рутину: обмер, реверс, проверки, развёртки, пакет производства.

Разделение труда

Кто	Что делает	Чем
Инженер	придумывает и рисует новую деталь, принимает решения по форме, радиусам, разгрузкам	FreeCAD — руками, привычно
Генератор кода	превращает нарисованное или реверснутое в код с параметрами; код открывается во FreeCAD как обычная деталь	recada.py → Python для FreeCAD
Инструмент	обмер, реверс, растяжение, операции, три критерия приёмки, DXF, BOM, пакет для Blehcon	headless-ядро на OCCT, веб-интерфейс
Редактор изделия	багажник как параметры: высота, положение, длина; ноги регенерируются	рабочее окно

Почему FreeCAD остаётся, а не уходит

Опыт показал: в цепочке автоматизации FreeCAD не нужен и даже мешает — его OCCT 7.8 не выполняет операции, которые проходят на 7.9 в нашем ядре. Но там, где нужна рука конструктора — новая форма, нестандартный узел, эскиз под конкретную машину — FreeCAD незаменим и бесплатен. Отсюда гибрид: FreeCAD рисует, инструмент считает и проверяет, генератор кода связывает их в обе стороны.

Идея workbench: как связать

Два пути, проверены обсуждением. Первый — полноценный верстак FreeCAD на его Part-API: требует переноса кода с OCP на другой API и упирается в отличия версий OCCT (offset у нас падал именно там). Второй — гибридный верстак: кнопки в FreeCAD сохраняют выделенное тело во временный STEP, вызывают внешний инструмент из его окружения и импортируют результат обратно. Ни переноса, ни конфликта версий; тяжёлая геометрия считается там, где она работает. Рекомендован второй путь. Делать его стоит после того, как устанутся операции ядра — оборачивать движущуюся цель в интерфейс рано.

VDI (Kasm/Guacamole) как способ дать внешним исполнителям FreeCAD в браузере остаётся в резерве — для сценария с недоверенными подрядчиками из раздела о найме. Для доверенной команды достаточно WireGuard и локального FreeCAD с генератором кода.

16. Инструмент реверса recada.py — построен и проверен

За два дня сентября собран рабочий инструмент, который снимает деталь с чужой геометрии и превращает её в проверенный производственный файл. Всё headless, без FreeCAD в цепочке: ядро OpenCASCADE подключено как библиотека (OCP), FreeCAD остаётся только смотрелкой. Один файл recada.py: командная строка и браузерный интерфейс.

Что умеет

Функция	Что делает	Проверено на
Обмер	толщина, гибы с двумя углами (поворот и двугранный), радиусы, отверстия с осями, разгрузки	Defender LRDE-2BR1: $t=3.0$, 7 гибов, 6 отверстий — повторяется до сотых
Расщепление на оболочки	виртуальный шуп: точка — нормаль $\times t$ попадает на поверхность $\rightarrow$ грань оболочечная; парование + двухцветная раскраска графа	41 грань покрывает 93 % площади; стороны 21/20
Реконструкция	кинематика на всех гранях тела, перестроение зоныгиба, заделка дыр	объём 78031.6 против эталонных 78031.6
Растяжение	разрез полупространством вдоль плоскости стенки, сдвиг, призма сечения; автоподбор места разреза по одиночному сечению	стенка +10 мм, +20 мм — замкнуто, годно
Операции	зеркало, сверление, заглушка, обрезка, скругления — с проверкой после каждой	все дают замкнутый солид
Развёртка	DXF с линиямигиба на отдельном слое	299 $\times$ 168 мм — совпало с порталом
Сборка	разбор STEP на детали с именами, группировка одинаковых, пакетный реверс	84.5 МБ, 628 объектов

Три критерия приёмки — главный вывод

Критериев оказалось не один и не два, а три, и третий не заменяется первыми. Это не теория: каждый найден через отказ.

Критерий	Чем проверяется	Что ловит	Как найден
Метрология	point-sampling двусторонний, расстояние до ПОВЕРХНОСТИ	точность геометрии	свой
Целостность	свободных рёбер = 0, солидов ≥ 1	дыры, незамкнутость	отказ портала Blehcon
Производство	загрузка на портал поставщика	распознаётся ли как листовая деталь	опыт

Три ошибки, найденные и исправленные

- Метрология мерила расстояние до тела, а не до поверхности. `distToShape` до солида возвращает ноль для точек внутри материала — «толстая» деталь проходила приёмку. Проверка: сдвиг детали на 0.3 мм давал 74 % точек в допуске, после исправления 29.8 %.
- Односторонняя выборка не видела лишний металл: кандидат длиннее эталона давал RMS 0 и 100 %. Невязка сделана двусторонней.
- Проверки замкнутости не было вовсе. Деталь может быть геометрически точной и дырявой одновременно — метрология этого не ловит. Обнаружено отказом портала, теперь считается одной строкой и предсказывает отказ до отправки.

Что не получилось и почему

Наращивание оболочки целиком (offset всей сшитой оболочки) оставляет дыры и зависит от версии OCCT: на 7.9 работает, на 7.8 внутри FreeCAD падает. Заменено поэлементным наращиванием с булевым объединением — для одной грани offset корректен всегда. Заплатки BRepFill в зонегиба дают сплайновую грань — распознаватель листового металла на такой ломается. Отсюда правило: в зонегиба строить только канонические поверхности (цилиндры

и плоские секторы), профиль разгрузки генерировать по правилу, а не латать.

Параметрический поворот угла гиба даёт замкнутое тело, но пока не проходит распознавание у поставщика: торцевой профиль гиба с разгрузкой надо генерировать заново, а не копировать. Это единственная незакрытая задача цепочки — она понятна до деталей.

17. Редактор изделия: багажник как параметр, а не как геометрия

Ключевая мысль, к которой пришли в работе: у багажника нет произвольных степеней свободы. У него есть продуктовые параметры, а положение и форма деталей — их следствия. Редактор поэтому не двигает детали, а регенерирует их.

Сценарий, ради которого всё делалось

Багажник спроектирован, между крышей и низом поперечины 20 мм — руку не просунуть. Решение: приподнять. Если ноги уже реверснуты в код, достаточно одного числа: точки крепления на крыше зафиксированы, вертикальная стенка каждой ноги растягивается на 20 мм, развёртки пересчитываются, DXF обновляются, три критерия приёмки прогоняются заново. Полный реверс всей системы не нужен.

Тот же цикл обслуживает аэродинамику: CFD говорит «сдвинуть платформу вперёд на 30, поднять на 10» — это два числа в `product.json`, а не неделя ручной работы. Именно поэтому гнутые детали обязаны жить как код с параметрами.

Единое рабочее окно

Зона	Содержимое
Дерево слева	таблица как BOM: видимость, выбор, имя, количество экземпляров, статус (исходник / реверснута / параметрическая), роль (нога, кронштейн крыши, платформа, поперечина, крепёж, кузов). Поиск по имени, автоопределение ролей по геометрии
Сцена	свой просмотрщик на <code>three.js</code> : разлёт, прозрачность, сечение, измерение, виды, изоляция. Выбранные подсвечены, реверснутые зелёные
Действия	для исходника — «Полный реверс» (одинаковые детали считаются один раз); для реверснутой — растяжение, отверстия, зеркало, DXF
Изделие	точки крепления к крыше как якорь, «поднять платформу на N мм» — регенерация всех ног
Журнал	<code>project.json</code> : каждая операция записана параметрами. Истина — журнал, геометрия — артефакт

Пакет производства

Одна кнопка собирает архив: `step/` — уникальные детали, `dxs/` — развёртки с линиямигиба, `passport/` — обмер каждой в JSON, `BOM.csv` — спецификация с количествами, `project.json` — журнал операций. Это и есть то, что уходит в Blechcon.

Что это меняет в экономике проекта

Реверс кронштейна Defender занял минуты вместо дней. Четыре одинаковые ноги считаются один раз. Подъём платформы — одно число вместо перепроектирования. Инфраструктура: EX44 за 59 евро нетто плюс Storage Box — около 63 евро в месяц за систему, которой нет в коробочных CAD: автоматический реверс листа с проверкой производственной пригодности.

18. Библиотека деталей и сервер: система с обратной связью

Сервер EX44 куплен, библиотека собственных STEP-файлов собирается. Вместе они превращают инструмент из демонстрации в систему, где каждая проверенная деталь делает следующую точнее.

Что даёт сервер

Разделение ядра и интерфейса: ядро — библиотека функций с командной строкой, интерфейс — тонкий слой над очередью задач. Реверс ставится в очередь, страница не блокируется, результаты сохраняются в постоянное хранилище по хешу файла. Без этого не заработают ни второй пользователь, ни библиотека. Непрерывная регрессия: каждый коммит в инструмент прогоняется по библиотеке ночью на CI, утром — одна цифра: сколько деталей проходят три критерия. Ломать работающее незаметно больше нельзя.

Общий доступ: удалённые инженеры из раздела о найме работают через WireGuard с одним инструментом и одними правилами.

Что даёт библиотека

Применение	Как работает	Что закрывает
Калибровка К-фактора	наша аналитическая развёртка против ответа портала Blehcon по каждой детали (299 × 168 подтверждено)	третий критерий из ручной проверки становится источником данных
Каталог разгрузок	типы, глубины, привязка к кромке — снятые с реальных деталей	незакрытую задачу зоны гиба: генерировать по правилу, не латать
DFM-правила	статистика радиусов, углов, толщин, диаметров, отступов от кромки	dfm_blehcon.json из практики, а не из головы
Поиск похожего	подпись детали: t, число гибов, набор углов, рисунок отверстий	взять проверенную и растянуть вместо проектирования с нуля
Паспорта авто	точки крепления и профили крыш как данные библиотеки	редактор изделия без ручного ввода координат
Стандарт школы	новый конструктор видит принятые радиусы и решения	передачу опыта без устных преданий

Граница честности: библиотека советует и предупреждает, решает человек. Она не скажет, выдержит ли деталь нагрузку, и не заменит инженерное решение там, где новая машина требует того, чего в практике ещё не было.

Рынок: Zoo как ориентир

Zoo (zoo.dev, бывший KittyCAD) — ближайший по философии проект: их язык KCL — читаемый текст как источник истины, версионирование в Git, агент Zookeeper переписывает STEP в код. Два отличия делают нашу нишу свободной: геометрический движок Zoo облачный и американский — 3D-вид это видеопоток с их серверов, чертежи уходят к ним; и у них нет производственной приёмки листового металла — метрологии, целостности, распознавания порталом поставщика. Zoo подтверждает направление и не закрывает нашу задачу.

Как Zoo стримит и что из этого следует

Движок Zoo работает на их серверах. Клиент ничего не считает: команды моделирования уходят по WebSocket, обратно приходят видеокadres по WebRTC — как видеозвонок. Стримится только 3D-вьюпорт; интерфейс, дерево и редактор кода живут в клиенте. Это не Kasm: Kasm стримит целый рабочий стол или приложение через захват экрана, тяжело и с задержкой; Zoo встроили рендер и кодек прямо в собственный движок на Rust, поэтому клиент лёгкий. Три причины их выбора: не портировать движок под платформы, открывать любые модели на слабых машинах, и главное — геометрия никогда не покидает их инфраструктуру, клиент видит только пиксели.

Для нас это независимое подтверждение принципа из раздела о VDI — «пиксели, не геометрия». Разница в реализации: они встроили стриминг в свой движок, мы стримили бы

готовое приложение. Обратная сторона нашей текущей схемы: просмотрщик на three.js отдаёт в браузер триангулированную сетку — для доверенной команды через WireGuard это нормально, для внешних подрядчиков это утечка, сетку можно сохранить. Поэтому для недоверенных остаётся стриминг пикселей.

Свой движок: ядро — нет, обвязка — да

«Движок» Zoo — три слоя: геометрическое ядро (B-Rep, булевы, скругления, развёртки), рендер и стриминг с API. Ядро они пишут годами командой специалистов по геометрическим ядрам с десятками миллионов инвестиций — и в марте 2026 всё ещё писали, что скругления рёбер остаются их болью. Это цена ядра. Нам его писать не нужно: OpenCASCADE — тридцать лет работы, бесплатно и открыто, и на нём вчера реверснута деталь, которую принял портал. Ядро — решённая задача чужими руками.

Что собрать реально можно — два верхних слоя вокруг OCCT. Рендер на сервере: наш three.js-просмотрщик в headless-браузере, либо встроенная визуализация OCCT. Стриминг: картинка по WebRTC в клиент, команды камеры обратно — готовые библиотеки, не исследование, порядок работы недели. Это «стриминг выюпорта вокруг OCCT», не «свой движок».

Когда это нужно: только для внешних исполнителей, которым нельзя отдавать геометрию. Для своей команды нынешняя схема проще и быстрее. Стриминг выюпорта — в резерве рядом с VDI, делается при появлении первого недоверенного подрядчика. Zoo пришлось писать ядро, потому что они продают платформу. Мы продаём багажники — разная экономика, разные решения.

Ядро напрямую: OpenCASCADE без FreeCAD

Точная формулировка архитектуры: OpenCASCADE и есть ядро внутри FreeCAD. Мы вызываем его напрямую через обёртку OCP, минуя FreeCAD — двигатель на стенде вместо автомобиля. Это даёт headless-работу на сервере и в CI, пакетную обработку библиотеки и свою версию ядра (7.9, где наращивание работает; в FreeCAD 7.8 оно падает). FreeCAD остаётся кабиной для инженера — там, где нужна рука. Одна геометрия, два способа обращения: программный для автоматики, ручной для человека. Практика: держать FreeCAD той сборки, где версия OCCT совпадает с ядром.

Ядро создано с ИИ, проверено реальностью, работает без ИИ. ИИ писал код ядра, но не работает внутри него: каждая функция — обычный детерминированный Python, один вход даёт один выход. Три ошибки ядра нашёл не ИИ — их нашли перепроверка, негативный тест и отказ портала. Отсюда правило разработки: тест раньше кода (Spec-Driven Development, применённый к самому инструменту), библиотека как регрессия после. Обучать свою модель не нужно и не выйдет — десятки деталей не выборка; библиотека работает как справочник. Готовый ИИ подключается как переводчик намерений к операциям ядра, не как источник геометрии.

Просмотрщики на том же ядре (состояние на сентябрь 2026)

Инструмент	Что это	Где применить
OCP.wasm + Yet Another CAD Viewer	OpenCascade и обёртка OCP скомпилированы в WebAssembly через Pyodide; build123d работает целиком в браузере. Та же OCP, что в recada.py	инструмент у клиента в браузере без сервера — ничего не покидает машину; ответ на утечку сетки для внешних подрядчиков
OpenCascade.js (ocjs.org)	порт OCCT в JavaScript/WASM через Emscripten, почти нативная скорость, многопоточность, урезанные сборки; на нём CascadeStudio	браузерный просмотр и лёгкие операции без утечки B-Rep
OCCT WebGL viewer	собственная визуализация OCCT в WASM — открывает BREP в браузере, нужен WebGL 2.0	показать «ровно как в ядре» без FreeCAD
Mayo	настольный просмотрщик и конвертер на Qt + OCCT: STEP, IGES, BREP, экспорт glTF; обновляется	бесплатный десктопный STEP-вьювер на нашем ядре — проверить, что видит ядро
OCP CAD Viewer (VS Code)	показывает объекты OCP/build123d прямо из Python, без STEP	отладка ядра при разработке

Инструмент	Что это	Где применить
Наш three.js-просмотрщик	сетка с сервера, выбор, разлёт, сечение, измерение	быстрый интерфейс для доверенной команды

Стратегически интереснее всего OCP.wasm: наша обёртка та же, значит recada.py может работать у клиента в браузере без сервера вообще — обратный подход Zoo, где всё улетает в облако. Для внешних исполнителей это третий путь наряду с VDI и стримингом выюпорта: считают у себя, отправляют только результат приёмки.

Версии ядра и формат на границах

OCCT 7.8 в FreeCAD 1.1, 7.9.3 в нашем ядре, 8.0 вышла в мае 2026 (крупнейший шаг за годы: чтение STEP до 75 % быстрее, потокобезопасное чтение и запись STEP, графовая топология BRepGraph, ускоренные экстремумы — всё под сервер и библиотеку; новых операций для листового металла нет). FreeCAD на восьмёрке — реалистично 2027 год. Правило: ядро инструмента фиксируется на одной версии, записывается в project.json, и переезжает только целиком — после прогона библиотеки-регрессии до и после. Ядро и FreeCAD никогда не обязаны совпадать; архитектура это переживает, потому что генератор кода вшивает готовые тела, а не пересчитывает их во FreeCAD.

Формат на границах — всегда STEP. BREP — внутренний формат ядра, его версия меняется с мажорным релизом, и 7.9 может не прочитать то, что записала 8. STEP — ISO 10303, читается любой версией в обе стороны, Inventor, SolidWorks, порталом поставщика. Правило в одну строку: BREP — внутри процесса, STEP — на любой границе: между инструментом и FreeCAD, между сервером и подрядчиком, между сегодняшним ядром и завтрашним. Цена — объём (800 КБ против 120 у BREP) — терпима.

STEP с именами: обязательное требование к экспорту

STEP хранит имена деталей и узлов, дерево сборки с положениями, количество экземпляров, цвета и слои — если писать через XCAF (STEPCAFControl_Writer с документом). На входе инструмент так и читает: имена Bracket_F_v2.1, wind_side_v2.1 приходят из сборки с положениями. На выходе сейчас пишет упрощённо (STEPControl_Writer) и имена теряет — дерево сплющивается в безымянный компаунд. Это ошибка инструмента, не формата; исправляется заменой писателя: имя, положение и цвет для каждой детали. Цветом кодируется статус: зелёный — реверснута.

STEP переносит	STEP не переносит — едет в project.json
имена деталей и узлов	обмер: толщина, гибы, отверстия
дерево сборки с положениями	статус реверса, роль
количество экземпляров	журнал операций
цвета и слои	точки крепления

Пара «STEP + project.json» — не избыточность, а разделение: STEP несёт то, что понимают все, JSON — то, что понимаем только мы; связь по имени детали. Безымянные экземпляры вида =>[0:1:1:5] в сборке — не потеря при чтении, а неназванные компоненты в исходном CAD; лечится именованием перед экспортом из Inventor.

Решение: каскадный выювер на сервере, наружу только пиксели

Принято после появления EX44. Ядро и рендер OCCT работают на сервере; визуализация OCCT (V3d, AIS) доступна из Python через ту же OCP — без C++ и без WASM. Браузер получает кадры и отправляет клики; сервер сам определяет, на какую грань, ребро или вершину пришёл клик — выбор точный по B-Rep, с привязкой к центрам отверстий и серединам рёбер. Геометрия не покидает сервер: ни сетка, ни STEP. Это модель Zoo на OCCT и раздел о VDI без Kasm.

Отвергнутая альтернатива — гибрид с локальным вращением по сетке: быстрее и дешевле для сервера, но сетка уходит в браузер и может быть сохранена. Для защиты данных неприемлемо.

Достаточно ли EX44

Видеокарты нет — рендер программный, через Mesa на процессоре. Сцена CAD проста: сотни тысяч треугольников, десятки кадров в секунду на ядро. Плюс кодирование кадра — ещё ядро. Итого два потока на активную сессию; активная — только пока человек вращает, при просмотре нагрузка нулевая. За вычетом обмера и инфраструктуры — пять-шесть одновременно вращающих. Для команды с запасом, для внешних подрядчиков — несколько человек одновременно.

Что расширяет запас без железа: во время вращения кадр пониженного разрешения, после остановки — один чёткий; лимит частоты кадров на сессию. Когда понадобится другое железо: при десяти и более одновременно активных — GEX-серверы Hetzner с GPU, цена в три-четыре раза выше; решение на потом.

Защита данных — не только стриминг

Мера	Что закрывает
Все библиотеки (three.js, рако и др.) на своём сервере	сейчас просмотрщик грузит их с внешних CDN — факт открытия детали виден третьей стороне по запросам
Доступ только через WireGuard, свой ключ у каждого	в логах видно, кто что открывал и когда
Без точек скачивания для внешних ролей	экспорт STEP и DXF — только доверенным
Водяной знак с именем сессии на кадре	снимок экрана запретить нельзя, атрибутировать — можно
Рендер по запросу, не поток	нагрузка на сервер и объём трафика

Граница, признанная заранее: от фотографии экрана не защищает ничто. Пиксели закрывают утечку системы — моделей целиком, параметров, библиотеки. Одну деталь перерисовать с экрана можно всегда: «перенабрать деталь можно, систему нельзя». Этапность: сначала статичный рендер по запросу — кадр с заданной камерой и номер грани по клику; непрерывный поток — позже, если понадобится. Место в очереди: после разделения ядра и интерфейса и очереди задач; строить точный выювер к инструменту, который меняется по три раза в день, рано.

Ответ на вопрос «как учить наш алгоритм» — на пальцах и как идея использования внешних ресурсов. Алгоритм не запоминает примеры и не угадывает: он работает по правилам и числам. Учить его — значит добавлять правила из отказов, уточнять числа по фактам и не давать забыть выученное. Внешние ресурсы — датасеты, портал поставщика — участвуют в этом как полигон и как оракул, не как учитель.

Три механизма обучения

Механизм	Как работает — на реальном примере	Куда пишется
1. Каждый отказ становится правилом	кронштейн с повёрнутым гибом отклонён порталом → причина: сплайновая поверхность в зонегиба → правило «в зонегиба только плоскости и цилиндры» → валидатор ловит за секунду. То же с дырами: отказ → «свободных рёбер ноль». Одна ситуация — одна строка; ошибка случается один раз	rules/dfm_blechcon.json, валидатор
2. Числа уточняются по фактам	К-фактор из справочника 0,42; портал считает развёртку 299 × 168, наша 298,4 → коэффициент для этой пары R/t сдвигается; двадцать деталей — таблица под реальный цех без единого вопроса поставщику. То же с пружинением, глубиной разгрузок, минимальной полкой	таблицы в rules/

Механизм	Как работает — на реальном примере	Куда пишется
3. Библиотека не даёт забыть	правка под новую деталь ломает старую незаметно (просмотрщик переделан шесть раз, каждый раз что-то отваливалось). После каждого изменения прогон по всем проверенным деталям: «было 47 из 50, стало 46» — что-то сломали, смотрим что	library/, регрессия в CI

Что не участвует: нейросеть. Ей нужны тысячи размеченных примеров и она ошибается вероятно; у нас десятки деталей и требование «либо годно, либо нет». Одной фразой: алгоритм учится не на примерах, а на ошибках, разобранных до причины.

Итог — не «больше правил»

Экспертные системы восьмидесятых умерли ровно от этого: тысячи правил, которые никто не мог ни проверить, ни отменить. Правил становится больше — но их держат в порядке три вещи. Правила обобщаются: три частных «глубина разгрузки при R3/t3 — 4, при R6/t3 — 5, при R3/t2 — 3» становятся одной формулой « $1,3 \times t + 0,2 \times R$ », которая короче, покрывает небывшие случаи и проверяема; настоящее обучение — когда список сжимается, а покрытие растёт. У каждого правила есть тест и счётчик: правило без детали в библиотеке, на которой оно срабатывает, — мёртвое; не сработавшее за год — кандидат на удаление; срабатывающее на каждой второй детали — сигнал, что оно слишком строгое или конструкторы систематически делают что-то не так. Правила лежат слоями: физика («нержавейка не гнётся с R меньше t» — вечное), профиль поставщика («матрицы 8, 10, 12, 16» — пока жив Blechcon), стандарт школы («у нас всегда R3» — меняется решением); разные владельцы, разная скорость изменений, не смешивать.

Реалистичный масштаб — десятки правил, не тысячи: листовой металл узок, BenDFM оценивает производительность десятком критериев; больше сотни — значит, что-то не обобщили. И главное: копят не правила, а данные. Библиотека с ответами портала растёт всегда; правила — её сжатая форма, пересчитываемая заново при смене поставщика или метода. Правила устаревают, данные — нет.

Правило и случай — две вещи, страхующие друг друга

Правило — обобщение: «отверстие не ближе 2,5 толщин к линиигиба»; применимо к любой детали, но слепо к контексту. Случай — конкретная деталь с исходом: «LRDE-2BR1, отверстие 8,5 в 6 мм отгиба, портал принял, в серии, рекламаций ноль»; ничего не обобщает, но это правда. На новой детали сначала правила — секунда, красное или зелёное; потом случаи — библиотека находит ближайшие по отпечатку и показывает прецедент: «три похожих прошли с первого раза, одна со второго, вот что правила». Где они спорят — самое ценное: правило говорит «слишком близко», а принятая деталь с тем же расстоянием лежит в библиотеке → либо правило смягчается с условием, либо случай помечается исключением. Без случаев правила черствеют, без правил случаи — архив. Цикл: новая деталь → правила → похожие случаи → решение человека → деталь становится новым случаем. Это рассуждение по прецедентам с производственным исходом вместо оценки «похоже».

Внешние ресурсы: тренировочная площадка, не учитель

BenDFM с двадцатью тысячами деталей не научит алгоритм ничему новому, но покажет, где он ошибается на формах, которых у нас нет. Нашли ошибку → поняли причину → правило. Учитель всё равно один: отказ, разобранный человеком. Роли внешних ресурсов по трём механизмам: для правил — синтетика и чужие формы (BenDFM, SMCAD, HeteroMF, ABC щупом) как источник ситуаций, в которых правило ещё не проверялось; для чисел — литературные таблицы (пружинение, BendGuide) и материальные карты (NUMISHEET 2025 для 1.4301) как стартовые значения до калибровки; для библиотеки — внешние наборы как регрессия на два порядка больше своей, чтобы ломать старое было видно на тысячах, а не на десятках. Раздел 42 содержит полный реестр с приоритетами.

Онлайн-калькулятор Blechcon как оракул

Портал возвращает на каждую деталь пять сигналов, а мы читали один: распознана ли как листовая, габарит развёртки, толщина и материал, цена по количеству, срок изготовления,

иногда замечания технолога.

Применение	Как	Что даёт
1. Приёмка	прошло / не прошло, развёртка совпала / нет	третий критерий — уже есть
2. Калибровка чисел	их развёртка против нашей по каждой паре R/t	K-фактор под их цех
3. Модель цены — самое ценное	спрашивать портал о вариантах, которых не собираемся заказывать: 2,5 против 3,0 мм; 1.4301 против 1.4404; партии 1, 10, 50; четырегиба против пяти — каждый ответ точка на кривой; 20–30 загрузок дают формулу $\text{цена} = f(\text{длина реза, гибы, материал, партия})$	«цена в контуре» (раздел 33) без писем — бесплатный оракул стоимости
4. Прецеденты	каждый ответ сохраняется к детали: принята, цена, срок, замечания	через год — история: как менялись цены, что стали и перестали принимать

Автоматизация: официального API у портала, насколько известно, нет — спросить. Через браузер технически несложно, но это чужой сервис: сотни автоматических загрузок за ночь могут нарушить условия и испортить отношения с поставщиком, которому вы нужны как клиент, а не как нагрузка. Порядок: сначала письмо — «хотим отправлять пачки вариантов для оценки; есть ли API или допустимо ли регулярно?». Клиент с готовыми файлами — лучший клиент цеха, обычно отвечают да. Чего не делать: гонять через портал чужие датасеты (BenDFM, ABC) — тысячи деталей, которые никогда не закажете; внешние наборы проверяются нашим инструментом, портал — только для своих деталей и их вариантов.

Надо ли знать оборудование Blehcon: обязательно — только то, что они публикуют (материалы, толщины, лист, радиусы, допуски): это огибающая возможностей, и портал уже проверяет против неё. Спросить один раз — стандарт инструмента, ширины матриц, усилие и длину стола: без этого три проверки (полка, тоннаж, коллизия) работают на «типичном» инструменте, с этим — на их; портал распознаёт лист и считает развёртку, но не симулирует гибку, коллизия на третьем шаге обнаружится у них на прессе. Не спрашивать — модели станков, внутренние таблицы, выбор последовательности оператором: их ремесло; всё это придёт косвенно через ответы. И зависимость: чем точнее валидатор под Blehcon, тем сильнее привязка — поэтому правила строятся как консервативное ядро под «любой нормальный цех» плюс профиль текущего поставщика (раздел 33, идея 9). Тогда знание о поставщике — преимущество, а не зависимость.

19. Инструкция по монтажу как артефакт изделия

Разобрана действующая инструкция Airholder для Defender 110 (Montageanleitung, 15 страниц). Вывод: около двух третей содержимого — прямые следствия product.json и BOM, треть — постоянные правила и юридика. Инструкция становится седьмым артефактом пакета производства рядом с STEP, DXF, паспортами, BOM, чертежами и project.json — с адресатом покупатель, а не Blechcon.

Что в инструкции откуда берётся

Страница / блок	Источник	Режим
Спецификация Komplettsset: 16 позиций с количествами (S13A ×6, B13A ×14, M8x14 DIN 912 ×56, Halterung 1/2 ×8+8...)	BOM; количество винтов — из числа соединений	авто
Пиктограммы деталей, виды сверху и изометрия, разнесённый вид	HLR-проекции с сервера в штриховом стиле	авто
Позиции сухарей в пазах: 14, 54, 44, 344, 15, 815 мм по профилям A/B/C/D	вычисляются из положений поперечин и кронштейнов вдоль профиля	авто
Масса багажника 48 кг в предупреждении о нагрузке	BOM с плотностями	авто
Шаги 1-13: соединение половин профиля D, рама, уголки, кронштейны, установка на крышу; выноски крепежа	порядок из графа связей; текст из шаблонов по типу соединения	шаблон + числа
Юридика, предупреждения, правило ориентации В-профиля, сборка на ровной поверхности, «правильно / неправильно» 10.1 и 10.2, два способа вставки сухаря	постоянные блоки — конструкторские знания, хранятся как правила изделия	ручной

Находка для модели изделия: «Completed connector» = Halterung 1 + Halterung 2. Нога состоит из двух гнутых деталей — кронштейн крыши и стойка — плюс резиновая подкладка и винт M6x22 в водосток. В редакторе изделия нога — узел из двух деталей с общим интерфейсом; подъём платформы растягивает только стойку, кронштейн крыши не трогает. Это подтверждает механику точек фиксации, реализованную в инструменте.

Инструменты

Задача	Инструмент	Статус
Штриховые виды как в инструкции	HLR из OCCT через OCP — проекции с видимыми и скрытыми линиями, вектор	ядро есть, модуль не задействован

Задача	Инструмент	Статус
Растр с тенями при необходимости	визуализация OCCT headless / three.js в безголовом браузере	резерв
Вектор в PDF	reportlab (собирает этот документ) + svgwrite	есть
Граф связей, порядок шагов	networkx + контакты деталей через BRepExtrema	есть
Шаблоны текста, DE/EN	Jinja2 + словари	добавить
Штрих-код EAN	python-barcode	добавить
Каталог крепежа DIN	своя таблица	добавить

Ключевой выбор — HLR вместо растра: инструкция сделана в штриховом стиле технической иллюстрации, HLR даёт ровно это в векторе для печати и считается на процессоре без графики.

Что добавить в систему — четыре слоя

1. Граф связей: какая деталь касается какой — из геометрии; из него порядок монтажа снизу вверх. 2. Модель соединений в product.json: пара деталей → крепёж, количество, позиция вдоль профиля, момент, фиксатор резьбы; из неё считаются 56 винтов, позиции сухарей и выноски — самый содержательный кусок, он же нужен для проверок зазоров и коллизий. 3. Серверный рендер по шаблонам видов: сверху, изометрия, разнесённый, шаг с подсветкой. 4. Генератор документа: шаблон страницы с рамкой, статичные блоки, подстановка данных, два языка. Сервер: HLR-проекция сборки — секунды на ядро; полная инструкция около полусотни картинок — минута-две; память — сотни мегабайт; GPU не нужен. Инструкция пересобирается на CI при каждом изменении изделия, как DXF. Что даёт сверх ручной работы: изменили число поперечин — пересчитались позиции сухарей, количества винтов, масса; Ranger получает инструкцию в день готовности геометрии; несоответствий между спецификацией и схемами не бывает по построению.

20. Два мира: B-Rep и сетка

Уточнение по практике: автомобиль в работе — не STEP, а FBX от сторонних моделеров, часто с неверным масштабом и пропорциями. Сейчас багажник ставится на машину в Blender: STEP багажника конвертируется в GLB, масштаб машины подгоняется вручную. Система делится на два мира с разной точностью, и смешивать их нельзя.

	Мир B-Rep	Мир сетки
Объект	багажник, детали, крепёж	автомобиль, окружение
Формат	STEP (на границах), BREP (внутри)	FBX на входе, GLB на границе
Точность	производственная: 0.05 мм, три критерия приёмки	визуальная: сантиметры, калибруется
Назначение	производство, реверс, параметрика	картинка, примерка, грубые зазоры, конфигуратор
Инструмент	recada.py на OCCT	Blender (скриптуемый, MCP уже есть)

Следствия

Точки крепления нельзя брать с модели машины — она неточная. Источник — собственные кронштейны, проверенные на реальной крыше: у Defender геометрия Halterung 1 и есть паспорт точек, снятый с металла. Логика разворачивается: не «машина задаёт багажник», а «багажник, который уже стоит, задаёт координаты машины». FBX подкладывается под паспорт для визуальной проверки.

Масштаб FBX калибруется по заводским размерам — колёсная база, ширина, высота крыши. Один коэффициент записывается в паспорт автомобиля вместе с моделью и переиспользуется вместо подгонки на глаз при каждой примерке.

Конвертация STEP → GLB без Blender: у OCCT есть собственный писатель glTF (RWGlTF_CafWriter, доступен из OCP), сохраняет имена деталей через XCAF. GLB багажника получается из инструмента headless на сервере, со структурой и именами.

Blender остаётся для того, что OCCT не умеет: чтение FBX, фотореалистичные рендеры для сайта, картинка машины в инструкции. Он скриптуем — примерка «взять GLB багажника, подставить откалиброванный FBX, отрендерить» автоматизируется через уже имеющийся Blender MCP.

Граница форматов: STEP — внутри B-Rep-мира и на границе с производством; GLB — на границе с визуализацией и конфигуратором Unity. Защита данных: сторонние FBX-модели машин часто лицензированы на одного пользователя и запрещают распространение — на сервере они лежат в той же защищённой зоне, что чертежи, наружу в конфигуратор уходят только рендеры или упрощённые сетки.

21. Данные: источники, каталоги, единая структура

По практике источников оказалось шесть, и им нужно одно место. Модель данных: два каталога — автомобили и аксессуары — с единой структурой «объект → представления с классом точности». Собственные детали в той же структуре: STEP, чертёж, ответ портала, паспорт обмера. Одна библиотека на всё.

Источники по убыванию точности и доверия

Источник	Что даёт	Ограничение
Aufbaurichtlinien производителя (VW Aufbaurichtlinien, Ford Body and Equipment Mounting Manual, Stellantis Conversion Guidelines)	что с кузовом разрешено делать: нагрузка на крышу статическая и динамическая, несущая структура (В-стойки, дуги крыши), зоны крепления с координатами, запретные зоны (подушки, датчики, проводка), центр тяжести, моменты и классы крепежа, антикоррозионные требования, точки электрики	доступ через порталы для Aufbaurhersteller после регистрации; версионирование по модельным годам
CAD от производителя (VW, Ford, Stellantis)	точная геометрия кузова, штатные точки, заводская система координат	NDA, лицензия; формат JT/CATIA требует конвертации (Inventor открывает JT) — один раз в STEP
2D-чертежи производителя	номинальные размеры и допуски штатных точек — что гарантируется для всех машин; единственная официальная геометрия для старых (Defender)	DXF читается ezdxf; PDF — вручную, чисел десятков
STL-скан	реальная машина или аксессуар как есть: допуски производства, накладки, модификации	регистрация в общую систему координат, чистка шума и дыр
Кронштейны в металле	подтверждённые точки крепления	только там, где уже стоим
CAD аксессуаров от поставщика	STEP, масса, интерфейс крепления	не все дают
FBX	картинка, конфигуратор	масштаб условный, калибруется по заводским размерам

Aufbaurichtlinien выше OEM-CAD по одной причине: CAD показывает геометрию, а Richtlinien — что с ней разрешено делать. Половина правил валидатора уровня изделия получает официальный источник с номером страницы вместо нашей оценки. Ссылка на Richtlinien в документации изделия — позиция в споре о гарантии и в сертификации: «кронштейн стоит в зоне, разрешённой производителем, нагрузка в пределах указанной». Для покупателя — сохранение заводской гарантии на кузов; ISO 11154 плюс соответствие Richtlinien — комплект, который дилер примет. Извлечение: готовый ИИ вытаскивает разделы о крыше, стойках, нагрузках и зонах в JSON с указанием страницы, инженер проверяет числа; каждое правило паспорта хранит документ, версию, страницу.

Единая структура: объект → представления

Представлен ие	Автомобиль	Аксессуар	Своя деталь	Роль
CAD	OEM	STEP поставщика	STEP из реверса или FreeCAD	номинал
2D-чертёж	заводской	схема отверстий	чертёж из HLR	номинал с допусками
Скан	реальная крыша	реальный тент, канистра	—	реальность
Визуальная модель	FBX	GLB	GLB из OCCT	картинка
Подтвержден ие	кронштейны в металле	—	ответ портала, паспорт обмера	факт

Не у каждого объекта есть все представления — у старого Defender нет OEM-CAD, у чужого тента нет STEP. Система работает с тем, что есть, и знает класс точности каждого источника.

Общая система координат — заводская

OEM-файлы приходят в стандартной системе автомобиля: X вдоль машины, ноль по производителю. Это и есть общая система, которой не хватало: скан выравнивается по ней (ICP, Open3D), FBX масштабируется по ней, паспорт записывается в ней. Там, где OEM-CAD нет, систему задаёт скан. Один раз выровняли — все проверки в одном пространстве.

Что открывают сканы — четвёртый критерий

Проверка посадки: грань кронштейна, которая ложится на крышу, сравнивается со сканом той же метрологией — точки грани → расстояние до сетки. Карта прилегания: где лежит, где висит, где упирается. Это критерий монтажный, не производственный; ни Blechcon, ни портал его не дают. Зазоры в реальности: поперечина к крыше, нога к рейлингу, платформа к антенне — от B-Rep до скана, без FBX. Сценарий «двадцать миллиметров, руку не просунуть» проверяется на настоящей крыше.

Скан против OEM-CAD — сама по себе информация: разбег у машин одной серии показывает, сколько регулировки должен иметь кронштейн. Продольные пазы Halterung 2 — ровно об этом.

Граница: реверс из скана в B-Rep — задача другого порядка, чем из STEP; для простого кронштейна реально, для сложной отливки нет. Скан аксессуара — источник размеров и проверок, не исходник для производства; адаптер под чужой аксессуар проектируется в FreeCAD по числам со скана.

Опция: реверс из STL

Статус — опция, не план. Отличается от реверса из STEP принципиально: в STEP плоскости и цилиндры уже заданы, в STL только треугольники — плоскость надо найти и подобрать, цилиндр гибок распознать по кривизне, отверстие восстановить по границе, каждый шаг с погрешностью скана.

Для листового металла реально: две поверхности — плоскости и цилиндры — лучший случай для распознавания (сегментация по нормальям, RANSAC, подбор цилиндра). Получаются те же числа, что даёт обмер: толщина по расстоянию между параллельными плоскостями, радиусы, углы, отверстия, контур. Ключевое решение: реверсировать не в B-Rep, а в параметры, а геометрию строить заново генератором — сшивать подобранные поверхности в тело не нужно, тело строит генератор, который уже проходит три критерия. Совпадает с принципом «истина — параметры».

Ограничения: качество скана (структурированный свет 0.05–0.1 мм, бытовые хуже; кромки скругляются, отверстия обрастают); лист снимать с двух сторон, иначе толщина — предположение; литьё, вытяжка, свободные поверхности — вне метода, там NURBS уровня Geomagic. Инструменты: Open3D, pyransac3d, scipy, далее наш генератор — всё открытое.

Где нужно: чужие детали без CAD; детали клиента под его крепления; и главное — входной контроль: скан детали от Blechcon против нашего STEP той же метрологией. Реверс наоборот — не восстанавливаем, а проверяем. Пятый критерий: приёмка поставки.

Аксессуары: вторая ось с данными

Интерфейс крепления описывается один раз для платформы (шаг пазов, сечение профиля) и один раз для аксессуара (в T-паз, на airline-рейку, на кронштейн, крепёж). Далее автоматически: аксессуар в конфигураторе показывается только там, где физически встанет; масса добавляется к нагрузке на крышу (нагрузка производителя минус багажник минус аксессуары); габарит проверяется на коллизии с кронштейнами и дефлектором; попадает в инструкцию отдельным шагом; тент на переднем краю — отдельный прогон CFD.

Инструменты для сеток: OCCT читает STL (RWStl); для тяжёлых сканов — trimesh для расстояний точка-сетка, Open3D для регистрации и чистки. Всё на процессоре; память под скан в 300 МБ — несколько гигабайт на время обработки, EX44 достаточно.

Защита: OEM-файлы под NDA — самое чувствительное на сервере; их утечка — нарушение договора с производителем. Пиксели наружу, личные ключи, запрет скачивания для внешних

— именно для них. OEM-геометрия никогда не уходит в конфигуратор и рендеры — там только багажники и FBX.

22. Конкуренты как источник данных

Конкуренты публикуют фото, PDF инструкций, размеры и массу. Всё это можно использовать — легально, пока анализируется, а не копируется геометрия. Конкурент — тот же объект библиотеки с представлениями низкой точности: инструкция как документ, извлечённая спецификация, габариты, масса, фото.

Что	Как	Что даёт
Структурный бенчмарк	из PDF инструкций: число деталей, крепёжных позиций, шагов, время монтажа, тип ног (одна или две детали), решение углов, задействованные точки крыши	таблица по конкурентам на одну машину: где проще, где сложнее
Подсказки для паспорта	инструкция показывает, куда конкурент крепится: водосток, рейлинг, штатные резьбы	подтверждение точек там, где OEM-данных нет
Метрики эффективности	масса на м ² платформы, высота над крышей, масса на точку крепления — в тех же единицах, что считает наша система из BOM	аргумент в карточку товара и цель для конструктора (55 мм против 70 у Front Runner)
Фото	фотограмметрия — грубая сетка	только габариты, пропорции, обтекаемость; не для реверса
Извлечение из PDF	готовый ИИ: PDF на входе, структурированный JSON на выходе, проверка глазами	спецификации конкурентов в базе, сравнения автоматически
Уроки DFM	как решают разгрузки, углы, стыки профилей	смотреть и учиться, не повторяя
Патентный поиск	по механизмам крепления на крышу до проектирования	недорогая страховка от претензии

Граница: публичные документы анализировать можно; воспроизводить геометрию, отличительные формы и узлы под патентом или зарегистрированным промышленным образцом — нет. Ценность — не повторить их багажник, а вывести свой быстрее и точнее.

23. Проверки уровня изделия: размещение, доступность, высота, настройка

Уровень изделия — не геометрия детали, а багажник на конкретной машине. Каждый вопрос «куда можно ставить», «достанет ли человек», «влезет ли в паркинг» превращается в правило валидатора: число, источник данных, порог. Каждая настройка — параметр product.json, каждый параметр тянет проверки заново.

Куда можно ставить

В паспорте машины — зоны. Разрешённые: водосток, рейлинги, штатные резьбы, с допустимой нагрузкой на точку от производителя. Запрещённые: люк, антенна, плавник, датчики систем помощи водителю, камера, стоп-сигнал, зона качания задней двери. Положение вдоль крыши проверяется против зон автоматически: сдвинули вперёд — валидатор сообщает, что передняя нога зашла в зону датчика дождя. Источники: OEM-CAD и чертежи для зон, скан для реальной машины, инструкции конкурентов как подтверждение точек.

Правила валидатора уровня изделия

Проверка	Что считается	Порог / источник
Зоны крепления	ноги и кронштейны внутри разрешённых зон, вне запрещённых	паспорт машины (OEM, чертежи, скан)
Нагрузка на крышу	производитель — масса багажника — аксессуары; —50 % на бездорожье	заводская допустимая нагрузка; из инструкции Airholder
Нагрузка на точку	распределение массы по точкам крепления	допустимое на точку от производителя
Двери и задняя дверь	дуга открывания против ног и заднего свеса платформы	кинематика двери из OEM или замер; полное открывание обязательно
Высота для паркинга	высота машины + багажник + самый высокий аксессуар	1.85 / 1.90 / 2.00 / 2.10 м; показывать в конфигураторе для каждой конфигурации
Загрузка с земли	высота нижней кромки платформы от земли	антропометрия DIN 33402; при превышении — ступенька из аксессуаров
Дотягиваемость	до центра платформы сбоку по ширине и высоте	DIN 33402
Зазор до крыши	низ поперечины ↔ крыша, нога ↔ рейлинг, платформа ↔ антенна	B-Rep ↔ скан или OEM; сценарий «20 мм, руку не просунуть»
Видимость	обзор через заднее стекло, камера заднего вида, антенна	луч обзора из OEM-положения зеркала
Аэродинамика	дефлектор, шумовые кромки, частоты срыва по Струхалю на поперечинах	формулы в валидаторе; CFD — разово
Центр тяжести	высота ЦТ машины с нагрузкой на крыше и аксессуарами	предел из Aufbaurichtlinien; для высокого фургона с палаткой — ограничивающий фактор
Динамика ISO 11154	город, экстренное торможение, боковой удар — расчётно	до испытаний в лаборатории
Совместимость аксессуаров	интерфейс, положение, масса, коллизии с кронштейнами и дефлектором	каталог аксессуаров

Каждое правило хранит источник со ссылкой — документ, версия, страница. Для зон крепления, нагрузки на крышу, нагрузки на точку, центра тяжести и крепежа источник — Aufbaurichtlinien производителя; для зазоров и посадки — скан и OEM-CAD; для доступности — DIN 33402; для динамики — ISO 11154. В конфигураторе покупателю показывается: «соответствует требованиям производителя к надстройкам».

Что регулируется без перепроектирования

Положение багажника вдоль крыши в пределах разрешённых зон; высота над крышей; шаг и число поперечин под груз; положение аксессуаров с проверкой совместимости; наличие и

ориентация дефлектора. Всё это параметры `product.json` — меняются числом, ноги и кронштейны регенерируются, проверки прогоняются заново, инструкция и DXF пересобираются.

Для покупателя в конфигураторе: полная высота с выбранными аксессуарами и вердикт по паркингу — до покупки, а не в паркинге. Это продолжение того, что уже есть в карточке T6 (2045 мм), только для каждой конфигурации автоматически.

24. Дизайн продукта: что даёт система

Всё предыдущее — про то, как деталь попадает в производство. Этот раздел — про то, каким должен быть сам багажник, и где система даёт конструктору не удобство, а новые решения. Три вещи опираются на данные, которые уже собираются: общность деталей между машинами, диапазон регулировки из разброса сканов, время монтажа из графа связей. Остальное — конструкторские правила, каждое кладётся в валидатор одной строкой.

Для покупателя

Время монтажа как проектный показатель. Сейчас 2–3 часа. Каждый шаг инструкции — минуты; сумма считается по графу связей. Система показывает, как предсобранные узлы её сокращают: нога в сборе с завода (Halterung 1 + 2 + подкладка, четыре винта) вместо двух деталей у клиента. Цель — час. Каждое изменение конструкции сразу видно во времени монтажа, а не после первой жалобы.

Один инструмент на всю сборку. Сейчас M8 и M6 — два ключа. Весь крепёж под один размер: инструкция короче, ошибок меньше, в коробку кладётся один ключ. Правило валидатора: число различных размеров крепежа в изделии — целевое значение 1, допустимое 2.

Сценарии груза. Палатка, велосипеды, каяк, канистры, ящики — у каждого своё требование к шагу поперечин, точкам крепления и распределению массы. Покупатель выбирает «что вожу», конфигуратор предлагает число поперечин, их позиции и аксессуары; валидатор проверяет нагрузку и высоту для паркинга с этим грузом. Сценарий груза — часть `product.json`.

Высота и паркинг — уже в разделе 23. Добавляется: показывать покупателю в конфигураторе полную высоту с выбранным грузом и аксессуарами, не только с пустым багажником.

Для конструктора

Семейство ног вместо ног под каждую машину — самое сильное, что даёт библиотека. Анализ общности деталей между машинами: если стойка Halterung 2 с регулировкой пазом покрывает и Defender, и Ranger, а различается только кронштейн крыши Halterung 1 — это в разы меньше артикулов, оснастки у Blechcon и складских остатков. Система считает общность автоматически, когда есть паспорта нескольких машин: для каждой детали — на скольких машинах она применима без изменений, с растяжением, с другим кронштейном. Целевой показатель: доля общих деталей в новом изделии.

Диапазон регулировки из разброса. Скан против OEM-CAD даёт, на сколько реальные машины отличаются от чертежа: допуски кузова, накладки, просадка. Длина паза в стойке и запас по высоте кронштейна должны покрывать этот разброс с коэффициентом — число, а не интуиция. Чем больше сканов одной модели, тем точнее диапазон. Правило: регулировка ≥ 2 $\times$ измеренный разброс.

Прочность и масса вместе. Планируемый CalculiX закрывает перебор толщины и формы ноги под нагрузку: где можно 2.5 вместо 3.0 мм, где нужны рёбра или отбортовка, где отверстия ослабляют. Масса на квадратный метр платформы — целевой показатель против конкурентов; модальный анализ — против дребезга на поперечинах.

Общая высота профиля как конкурентное поле: 55 мм у Airholder на T6 против 70 у Front Runner. Держать ниже конкурента — правило валидатора для каждой новой машины.

Для эксплуатации

Гальваническая пара. Нержавейка 1.4301 на алюминии в мокрой среде с солью — коррозия на стыке через сезон. Резиновые подкладки Gummiunterlage у вас есть; правило валидатора: между разнородными металлами всегда изолятор, и он обязан присутствовать в BOM на каждом контакте. Проверяется по графу связей: контакт сталь-алюминий без изолятора — ошибка.

Расходники и запчасти. Подкладки, заглушки, крепёж изнашиваются и теряются. BOM с именами и номерами позиций из инструкции делает возможным заказ отдельной детали по номеру — сервисная выручка, которой сейчас нет, и удержание клиента. Каждая позиция BOM

— потенциальный артикул в Shopify.

Защита от кражи — гайки с секретом на кронштейнах крыши как аксессуар: спрашивают, стоит копейки, повышает средний чек.

Обслуживание после бездорожья: инструкция содержит проверку затяжки; система может выдавать карту контрольных точек — какие винты проверить и с каким моментом — из той же модели соединений.

Для бренда и прослеживаемости

Серийный номер и QR-код на изделии: ведёт на инструкцию именно этой конфигурации, регистрацию гарантии, заказ запчастей. Прослеживаемость по ISO 9001 из коробки: серийник → product.json → партия у Blehcon → ответ портала. Отзыв партии при дефекте — по базе, а не по памяти.

Фирменные элементы как параметрика: логотип уже вырезается лазером — деталь U5-Logo в сборке. Логотип, форма углов, рисунок дефлектора ведутся в той же системе и регенерируются под размер изделия, сохраняя узнаваемость.

Разборность и переработка: алюминий и нержавейка разделяются без инструмента — для немецкого рынка заявляемое преимущество; система может считать долю материалов по BOM и подтверждать разделяемость по графу связей (нет клеёных и неразъёмных соединений разнородных металлов).

Что из этого системное

Решение	На чём основано	Показатель
Семейство ног	паспорта нескольких машин, библиотека	доля общих деталей в новом изделии
Диапазон регулировки	скан против OEM, разброс по машинам	регулировка $\geq 2 \times$ разброс
Время монтажа	граф связей, минуты на шаг	цель 1 час; предсборка узлов
Один инструмент	BOM	размеров крепежа: 1, допустимо 2
Гальваническая изоляция	граф связей + материалы	контактов сталь-алюминий без изолятора: 0
Масса и прочность	BOM + CalculiX	кг/м ² ниже конкурента; запас по ISO 11154
Высота профиля	геометрия	ниже конкурента на той же машине
Прослеживаемость	серийник → product.json → партия → портал	отзыв по базе

Общий принцип: каждое дизайнерское решение, которое можно выразить числом, становится правилом валидатора и целевым показателем. Тогда следующая машина проектируется не «как получится», а против измеримых целей, и отклонение видно до производства.

25. Обратная связь из поля: рекламации, опыт покупателей, соцсети

До сих пор обратная связь шла от портала и от испытаний. Этот раздел — про сторону после продажи: от дороги и людей. Три источника, у каждого своя механика; все они — ещё два представления в библиотеке у изделия и у машины: «полевые данные». Цикл тот же, что с порталом: поле → библиотека → правило → валидатор → следующая ревизия product.json.

Рекламации — как ответы портала, только про эксплуатацию

Каждая жалоба — структурированная запись: машина, конфигурация по серийному номеру, деталь, вид отказа (ослаб винт, коррозия на стыке, гул на скорости, не встало на крышу), фото. Серийник ведёт к product.json, к партии у Blehcon, к ответу портала. Отсюда: частота отказов по детали и по машине; корневая причина по цепочке — конструкция, партия или монтаж клиентом; правило в валидатор: три жалобы на ослабление одного винта — в инструкцию момент затяжки и фиксатор, в модель соединений — требование. Ровно как отказ портала стал критерием приёмки. Это CAPA-цикл ISO 9001 с данными вместо папки; стоимость отказа по видам определяет, что чинить первым.

Опыт покупателей — прямо из инструкции

QR-код на каждом шаге: «шаг непонятен», «заняло дольше», «деталь не подошла». Время монтажа, ставшее проектным показателем в 24, измеряется реальными людьми, а не оценкой по графу. Шаг, на котором спотыкаются все, переделывается: предсборка узла или другая картинка.

Соцсети — то, о чём не пишут в жалобах

Что показывают	Что это значит	Куда идёт
Самодельные крепления канистр, фонарей, лопат	спрос на аксессуары, которых нет	каталог аксессуаров, приоритет разработки
Где сверлят платформу	штатного интерфейса не хватило; каждое чужое отверстие — недостающая позиция паза или аксессуар	интерфейс платформы, каталог
Как грузят	реальные сценарии груза и распределение массы	сценарии груза в product.json (15и)
На каких машинах	багажник на машине, для которой не делали — спрос на программу	очередь вывода машин
Что ломается и как чинят	рекламации, которые не дошли	библиотека отказов, валидатор

Механика сбора: посты по хэштегам и упоминаниям; готовый ИИ читает и раскладывает по меткам — аксессуар, машина, модификация, отказ, сценарий. Не аналитика больших данных, а еженедельная сводка: «12 постов с самодельным креплением лопаты за месяц». Плюс материал для маркетинга с разрешения автора.

Шестой критерий — полевой

Критерии приёмки выстраиваются в цепочку: метрология, целостность, распознавание порталом — производство; посадка по скану — монтаж; входной контроль поставки; и полевой — изделие приняли не Blehcon и не лаборатория, а дорога и люди. По серийникам видно, какая ревизия стоит у кого; отзыв и улучшение — адресные, а не всей серии.

26. Модель версий: уроки Onshape

Источник — статья «Under the Hood: How Collaboration Works» Ильи Барана, VP по архитектуре Onshape (единственный CAD с Git-подобным ветвлением). В ней описан принцип, к которому мы пришли сами — «истина — определение, геометрия — артефакт», — доведённый до конца за десять лет. Onshape как продукт нам не подходит (облако PTC в США), но модель данных берём.

Как устроено у Onshape

Три типа данных разделены жёстко: состояние интерфейса не хранится; определение — список операций, имена, импорт — единственное, что лежит в базе; результаты перестроения — B-Rep, треугольники, ошибки — кэшируются и всегда перестраиваются из определения.

Микроверсия: определение — вечный неизменяемый объект; любое изменение создаёт новую микроверсию со ссылкой на родителя и самим изменением («глубина Extrude 1 → 4 дюйма»). Вся эволюция хранится, любое состояние восстанавливается.

Устойчивые изменения — ключевая находка: изменение адресует операцию по внутреннему постоянному идентификатору, не по позиции; пока исходная операция существует, изменение применимо к другой микроверсии. Поэтому правки нескольких людей ложатся на последнюю версию, не мешая друг другу.

Ветвление: рабочее пространство — указатель на последнюю микроверсию; микроверсии образуют дерево. Слияние — найти изменения из «откуда», которых нет в «куда», создать микроверсию с ними. Сознательное отличие от Git: конфликтов не бывает — при разном изменении одного размера побеждает «откуда», пользователь никогда не блокируется. Отмена — не откат, а применение обратного изменения поверх последней версии: моё отверстие исчезает, чужое скругление остаётся. Compare — сравнение версий на тех же неизменяемых микроверсиях.

Что берём

У Onshape	У нас	Статус
определение / результат	project.json / STEP-артефакты	есть; формализовать: STEP — кэш, не истина; в Git только для релизов
микроверсия с родителем и дельтой	плоский журнал	сделать: каждая операция — объект с хешем родителя
адресация по постоянному ID	отверстия по центру, стенка по нормали и площади	частично; довести до правила для всех операций
переигрывание на новом родителе	нет	главное: журнал применяется к новой ревизии исходника — Ranger-кронштейн v2 получает те же правки
ветки и слияние без конфликтов	нет	на уровне журнала, правило «источник побеждает»
сравнение версий	нет	геометрический diff: грани добавлены, удалены, сдвинуты — подсветкой
тесселяция по требованию, кэш	кэш Streamlit	формализовать: кэш по хешу микроверсии
FeatureScript	генератор на Python/OCCT	подтверждение абстракции; ничего менять

Где у нас сложнее

У Onshape своё ядро, и оно ведёт постоянные идентификаторы граней через перестроения. У OCCT их нет — это известная проблема топологического именования, на которой FreeCAD бился годами. Мы адресуем элементы геометрическим отпечатком: тип, нормаль, центр, площадь. Для листовых деталей это работает — так уже находятся отверстия и стенки. Универсально — нет; это признано. Ещё две UX-идеи Onshape для нашего просмотрщика:

синхронный показ развёртки и согнутой детали (клик по грани в одном окне подсвечивает её в другом) и визуальное сравнение ревизий детали.

27. Инфраструктура: решения по железу

Почему Германия — самые дешёвые серверы мира: промышленные энергоконтракты + free cooling, DE-CIX (почти бесплатный трафик → unlimited), ЦОДы в дешёвой провинции при толстой оптике, культура ценовой конкуренции (Hetzner/netcup — семейные компании без инвесторов), вертикальная интеграция с десктопным железом. AWS отвергнут: 6–8× цена инстанса, egress ~0,09 \$/ГБ убивает VDI-поток (у Hetzner unlimited), CLOUD Act против data sovereignty, сложность без пользы. Совместимость бесплатна (Docker переезжает за день), зависимость не нужна.

Роль	Решение	Статус/цена
Целевой сервер (ядро+VDI 8-10 сессий)	Hetzner EX44: i5-13500 (быстрый однопоток — ключ к отзывчивости), 64 ГБ, 2×512 NVMe RAID 1, unlimited, FSN1	€70,21, setup 0, почасово; Limited-волны — подписка на availability, правило 15 секунд на заказ
Параллельный канал	Server Auction (Robot → Server Market): запросы «i5-13», «i5-12», «i7-12»; NVMe, не HDD; ничего старше 12-го поколения	критерий: ≤ €75–80
План Б (через ~3 недели)	OVH RISE-S: Ryzen 9700X (лучший однопоток), 64 ГБ, Frankfurt/Strasbourg, месяц-к-месяцу	€78 + €78 setup
Отвергнуто	AX41 (Zen 2 2019 за те же €70), AX42 (€118 за 35-Вт чип), EX63 (€177 — переплата за ядра), i9-13900 аукцион (€165 — рано), Contabo (CPU steal)	—
Масштаб (этап 5, 10+ сессий)	OVH RISE-L (9950X, 128 ГБ, €179) — вытеснил AX102 (€308); i9 с аукциона	по факту роста
Уже есть	ThinkPad i7-8850H/32ГБ/Quadro P1000 — рабочее место + локальный старт этапа 1 (Git, build123d, FreeCAD; диск 54 ГБ свободно — Docker позже); Strato VPS 2/2/90 — Gitea+WireGuard и/или вторая копия бэкапа; Hetzner CPX — CI	0 € новых
Второй провайдер	netcup: RS G12 (выделенные EPYC-ядра, ECC, дешево) — бэкап сейчас (~5 €), в будущем возможное отдельное ядро (Gitea+CI) при VDI на десктопном CPU; EPYC-однопоток слаб для VDI	~5 €/мес

Три входа Hetzner: accounts (паспорт) → Cloud Console (облачные CPX) и Robot (dedicated: заказ, rescue, Server Market). EX44 после поставки: rescue → installimage → Ubuntu 24.04 LTS + RAID 1 → SSH-ключ, ufw (наружу только SSH+WireGuard), unattended-upgrades, бэкап в Storage Box (~4 €) + вторая копия у netcup. Домен: поддомены git./work./cdn. существующего домена (или отдельный технический ~15 €/год) — Let's Encrypt, стабильность при переездах, разделение контуров; рабочие поддомены могут резолвиться только внутри WireGuard. Авторизация — три непересекающихся контура: рабочий (локальные аккаунты Gitea+Kasm + TOTP 2FA + WireGuard-периметр; SSO/Keycloak после 20 человек), клиентский (Shopify Customer Accounts / Google — конфигуратор наследует), сообщество (OAuth GitHub/Google в публичной Gitea). Бюджет полной системы при 100 зарегистрированных (пик 20–25 сессий): ~150–200 €/мес.

Стоимость владения: только open source

Категория	Инструменты (все бесплатны, без подписок)
CAD и генерация	FreeCAD + SheetMetal + аддоны (GPL/LGPL), build123d/CadQuery (Apache)
Версии и сервер	Git, Gitea, Git LFS, zippey; Ubuntu, Docker (self-hosted), WireGuard, ufw; Let's Encrypt; Cloudflare free (CDN)
Расчёты	CalculiX + Gmsh (статика, модальный), OpenFOAM (CFD разово)
Рендер и артефакты	Blender headless, SheetNest (раскрой), gltfpack/Draco (GLB)
VDI	Kasm Community Edition (до 5 одновременных сессий); без лимитов — Apache Guacamole / webtop-образы linuxserver.io

Категория	Инструменты (все бесплатны, без подписок)
Визуализация	Mermaid/Graphviz (граф зависимостей), React Flow/Rete.js (composep, этап 5), Sverchok (нодовое моделирование)
Конфигуратор	Unity Personal — бесплатен до \$200k годовой выручки с продукта; единственный инструмент с порогом; бесплатная альтернатива при необходимости — three.js/Babylon.js

Платные строки проекта — только три, все приняты осознанно: серверы (~70-200 €/мес по росту), токены Claude/API (~20-50 €/мес на старте), физический мир (FDM-принтер разово 300-500 €; испытания ISO 11154 — разовые тысячи €). Правило проекта: инструмент входит в стек, только если он open source или бесплатен для нашего масштаба без подписки; платными могут быть только вычислительные ресурсы и физические вещи; кандидат на платный инструмент требует явного решения владельца с бесплатной альтернативой в кармане.

Claude как копилот — четыре слоя: (1) Claude Code на сервере — админ-диалог вместо ручного Linux; (2) MCP-FreeCAD headless — двигатель промпт-слоя; (3) в CI неинтерактивно (claude -p): упал валидатор → фикс коммитом в ветку → ревью человеком; перевод сданных STEP в черновики генераторов; (4) в VDI-образе подрядчика — отдельный API-ключ с лимитом и системный промпт RECADA (ядро невидимо). Правило: Claude ходит через те же двери, что люди (ветки, PR, валидатор), автор коммита «LLM via MCP» с промптом в сообщении. Бюджет 20-50 €/мес на старте.

28. Граф зависимостей конструкции и автоматизация

Граф зависимостей — визуальная карта конструкции: узлы — файлы истины (паспорта, скелеты, генераторы, детали, сборки, конфигурации), стрелки — «читает из». Зависимости в системе объявлены явно, поэтому карта строится автоматически скриптом CI (Mermaid/Graphviz, бесплатно) — рисовать руками ничего не надо, схема всегда актуальна. Главный режим — анализ изменения: перед merge CI отвечает «что пересоберётся, если принять этот PR» — список и подсветка на схеме (красное = затронуто, отдельно помечаются released-детали). Каскад «что за что тянет» становится видимым ДО нажатия кнопки; это же — инструмент онбординга (новый человек понимает систему за час по карте) и диагностики (где в цепочке красный валидатор). Делается на этапе 2 рядом с валидатором.

Автоматизация бизнес-событий — простыми скриптами и webhook в CI, без отдельного оркестратора: складской webhook Shopify (остаток A=0 → политика run-out → активировать Б → уведомление), онбординг (тест пройден → карточка исполнителя → письмо), мониторинг (релизный тег → сводка в канал; CI красный сутки → алерт), маркетинг (релиз → рендер → черновик карточки через LLM на утверждение), продажи как источник ТЗ (форма «нет крепления?» → issue + счётчик спроса → 3 запроса = приоритет в разработку). Граница неизменна: автоматизация не трогает геометрию и не пишет в main — всё, что меняет истину, идёт через PR и валидатор. Скрипт создаётся под событие, случившееся руками трижды. Нодовый composer поверх библиотеки кубиков (React Flow/Rete.js — открытые) — этап 5: интерфейс для не-программистов (дилер-CPQ, клиент с шаблоном); свой Grasshopper не делать (есть Sverchok/FreeCAD-ноды — бесплатные).

29. Финансовая модель — каркас

Поставщики и данные для честной экономики: структурированный архив счетов Blechcon и тарифа GLS с первого заказа + pre-flight nesting (знать справедливую цену резки до запроса) + периодический ручной бенчмарк 1-2 европейских альтернатив + квалификация второго поставщика как страховки от монополии. Полный API-аукцион нескольких сервисов отвергнут: правила валидатора = правила конкретного производителя, несколько поставщиков = кратная поддержка DFM-правил.

CM1 (маржа комплекта) = Цена – металл – Blechcon – профили – крепёж/покрытие – упаковка/доставка – комиссия канала CM2 (с учётом дизайна) = CM1 × ожидаемые продажи модели – (турниры + часы владельца × ставка + прототипы + паспорт) Операционная = Σ CM2 – постоянные (инфраструктура, время, маркетинг, склад) Ориентир отрасли (4x4-аксессуары): валовая 45–60% в рознице, 30–40% через дилеров; ниже ~35–40% от розницы — чинить до масштабирования. Ключевой вопрос системы: за сколько проданных комплектов окупается вывод новой машины. Для расчёта нужны 8 цифр: розничная цена комплекта и канал; счёт Blechcon за комплект ног (или масса); €/метр профиля и метраж; крепёж/уплотнители/покрытие; упаковка+доставка; текущие и плановые объёмы; внутренняя ставка часа владельца; текущая стоимость вывода машины по-старому. Метрика №2 включает доставку металла — юнит-экономика честная.

30. Управленческий слой: роли подробно

Роли — это не должности, а зоны ответственности с чёткими правами решений; в малой команде один человек совмещает несколько ролей, но границы решений остаются письменными. Принцип: у каждого решения — один владелец; «решили вместе» = не решил никто.

Роль	За что отвечает	Права решений	Доступы	Время
Owner (владелец)	Стратегия и деньги; приоритеты трубы (что делаем следующим); критерии приёмки; доверие людям; отношения с Blechcon и ключевыми партнёрами; развитие системы (ворота этапов)	Единолично: релизная кнопка (tag), бюджеты трёх конвертов, повышения по лестнице, изменение интерфейса платформы, изменение released-деталей (go на issue), допуск инструментов в стек (FOSS-правило)	Полный: Git-admin, сервер root, Robot/Cloud Hetzner, Shopify, банк	Цель — сжатие до управления трубой; всё, что затягивает внутрь (рисовать, чинить CI), — сигнал незакрытой роли
Lead Reviewer	Первичная приёмка всех PR (геометрия, посадка, собираемость, экономика по BOM, эстетика); ведение исполнителей (ответы на вопросы к ТЗ, разбор отказов валидатора); формулировка ТЗ вместе с Owner; наставничество новичков; предложение правил в rules/ из повторяющихся ошибок	Merge в main для деталей (кроме изменений интерфейса и released — эскалация Owner); отклонение работ; распределение задач по скамейке; эскалация повышений	Git: merge-права на products/accessories; чтение generators и rules; Gitea-канбан; без root сервера	Первая делегируемая роль — решает bus factor: минимум раз в месяц проводит приёмку полностью без Owner. Отбор — по качеству мышления в комментариях, не по скорости черчения (принцип Питера)
Senior Engineer	Сложные узлы и новые типы деталей (топология); перевод «песочница → код» (реверс в генераторы); развитие библиотеки кубиков; калибровка FEM по эталонным отчётам; сложные паспорта (windshield-поверхности)	Коммиты в generators/ через PR; создание веток convert/*; предложение новых кубиков; техрешения внутри ТЗ без согласования	Git-доступ (режим Б: локальная работа + push), выбранные репозитории; VDI по желанию; без прав merge в main	Вырастает из проверенных по логам (устойчивое качество + коммуникация); финальное доверие — личное решение Owner
Engineer (проверенный)	Прямые задачи по готовым шаблонам и ТЗ: адаптеры, вариации ног, паспорта аксессуаров по чек-листу; печатные примерки своих деталей	Работа в своей ветке; сдача через PR; вопросы к ТЗ приветствуются (логируются как сигнал роста)	VDI-сессия с расширенным набором репозитория; либо локально + Git по решению Owner	3+ принятых работы; сдельная оплата выше новичка; очередь на задачи

Роль	За что отвечает	Права решений	Доступы	Время
Engineer (новичок)	Тестовые и простые задачи строго по ТЗ; работа только в изолированной VDI-сессии	Только своя ветка своей задачи; ничего кроме	VDI-сессия: один репозиторий задачи, egress только Gitea, водяной знак, без скачивания	Вход свободный, платный тест с автоприёмкой; риск компании — сотни евро
Ops (инфраструктура)	Сервер, Docker, Gitea, Kasm, WireGuard, бэкапы (+ квартальный тест восстановления!), обновления с фиксацией версий, мониторинг, онбординг/оффбординг аккаунтов (одна процедура: ключи, сессии, права)	Плановые обновления и правки конфигурации; экстренные действия при инцидентах с пост-отчётом Owner; НЕ решает про доступы людей (это Owner)	Root сервера, admin Gitea/Kasm; без прав в конструкторские репозитории по содержанию	10–20% времени одного человека; на старте совмещает Owner (+Claude Code как копилот), позже — первый кандидат на ау тсорс/частичную занятость
Технолог Blechcon (внешняя)	Производственное ревью выпускаемых деталей («если гнуть иначе — на 15% дешевле»); консультации по граничным случаям DFM	Рекомендательные; замечания конвертируются в правила rules/ через PR (цикл CAPA)	Получает только конкретные DXF/STEP по заказу; доступа в систему не имеет	По договорённости, возможно платно; дешёвый канал чужого производственного опыта

Правила совмещения и роста

На старте: Owner совмещает Lead Reviewer и Ops (с Claude как копилотом обеих ролей), Senior — по мере появления. Порядок делегирования: сначала Lead Reviewer (снимает приёмку — главный пожиратель времени Owner и единственное лекарство от bus factor), затем Ops, затем постановка ТЗ. Несовместимость: автор детали не может быть её единственным ревьюером — подписать на released всегда чужая; Ops не решает про доверие и доступы людей. Каждое повышение = расширение доступов по гибридной модели (VDI-only → +репозитории → Git локально → merge-права) и фиксируется в карточке исполнителя в Gitea. Оффбординг любой роли — одна процедура: отзыв WireGuard-ключа, закрытие сессий, ревизия прав, чек передачи незавершённого.

Ритм и бюджет (без изменений)

Релизный такт 2 недели; еженедельный 30-мин обзор трубы (канбан); месячный бизнес-срез (юнит-экономика, метрики); квартальные ворота со стоп-лоссами. Бюджет — три конверта: Infra (~50–150 €/мес фикс), Design (N деталей × целевая стоимость), System-dev (только под ворота); часы Owner — всегда отдельной строкой по внутренней ставке.

№	Метрика	Смысл
1	Время «задача → released»	скорость трубы
2	Полная стоимость released-детали (исполнители + часы владельца + прототипы + доставка металла)	честная цена дизайна
3	Доля прохода валидатора с 1-го раза	качество пула и ТЗ
4	Время добавления новой машины	главная метрика системы
5	Доля выручки от деталей без участия владельца в рисовании	освобождённость владельца

31. Материалы и технологии

32. Внешние ресурсы: что берём и чего не берём

33. Скорость: что ещё ускоряет вывод багажника

34. Удалённые столы: способы, выбор, география

Сервер EX44 есть; удалённый инженер работает на нём, данные не покидают сервер. Выбор способа зависит от того, что отдаётся: весь стол, одно приложение или только 3D-окно.

Способ	Что это	Плюс	Минус
xrdp	Linux-стол по RDP, клиент встроен в Windows	самый экономный, терпим к задержке, нулевая настройка у клиента	3D идёт картинками, вращение вязкое
Guacamole (Apache)	шлюз RDP/VNC/SSH через браузер	открытый, бесплатный без ограничений, запись сессий, 2FA, запрет буфера и файлов	сам ничего не запускает — нужен стол
Kasm Workspaces	стол в одноразовом контейнере на пользователя, в браузере	полная изоляция, водяные знаки, запреты встроены	тяжёлый; бесплатен до 5, дальше лицензия
Selkies	WebRTC-стол в браузере, открытый	видеопоток, плавное 3D	настройка, кодирование ест CPU
Xpra	одно приложение как отдельное окно у клиента	экономнее полного стола	—
NoMachine	протокол NX	очень эффективный	закрытый, лицензия для компании
Sunshine + Moonlight	игровой стриминг	минимальная задержка	под GPU; без него грузит CPU
Стриминг вьюпорта ОССТ	только 3D-окно (15в)	легче и защищённее всех	только просмотр и выбор

Guacamole против Kasm: Guacamole — дверь к столу, Kasm — стол в контейнере. Для доверенной команды до пяти человек — Guacamole с xrdp: проще, легче, бесплатно. Для недоверенных подрядчиков — Kasm: изоляция по контейнеру. Не исключают друг друга; Kasm в резерве до первого внешнего исполнителя.

Рекомендуемая схема: WireGuard (личный ключ) → Guacamole (2FA, запись, буфер выключен) → xrdp-стол с FreeCAD, инструментом и Blender. Всё открытое, всё на EX44, стоимость ноль. Для интерактивного Blender с тяжёлыми сценами без GPU тесно — рендеры пакетно ночью.

География и задержка

Скорость определяет физика, не софт: Восточная Европа и Турция 30–50 мс, Северная Африка и Ближний Восток 60–90, Индия 130–180, Юго-Восточная Азия 180–220, Латинская Америка 200–250. До 80 мс — как местная работа; 100–150 терпимо; выше 150 — каждый клик с запозданием, 3D мучительно. Софт вносит единицы миллисекунд; протокол — заметно: RDP и Xpra передают команды интерфейса и терпят задержку, Kasm и Selkies стримят видео и на далёком канале мутнеют. Для дальних стран — Guacamole с RDP. Важнее обоих — канал подрядчика: проводной от 10 Мбит с ровной задержкой, проверять тестом до найма. Второй сервер ближе к подрядчику не поможет: данные в Германии, переносить их — та утечка, которую исключили.

Комфортная зона по задержке — Восточная Европа, Балканы, Турция, Кавказ, Северная Африка: любой протокол, полноценная работа с 3D. Индия и Юго-Восточная Азия — для задач без активного вращения: параметры, проверки, библиотека, инструкции. Совпадает с раскладом ставок раздела 9.

35. Найм: три профиля и тесты самой системой

Система меняет профиль того, кого искать: форму задаёт владелец, инструмент считает, проверяет и упаковывает. Три уровня — три разных найма.

Профиль	Что делает	Уровень	Где и как
Инженер-оператор системы	гоняет детали через реверс и портал, ведёт библиотеку и паспорта, переносит Aufbaurichtlinien в JSON, собирает пакеты для Blechcon, правит инструкции	младший или средний по листовому металлу, знающий гибку руками	постоянный, география до 80 мс; недорог — правила в валидаторе, не в голове
Конструктор аксессуаров	создаёт то, чего нет: интерфейс между чужим объектом (бокс, тент) и платформой; варианты концепции с компромиссами; нагрузки, вибрация, ветер, доступ руками	средний или старший с выпущенными в серию изделиями (aftermarket, outdoor, крепёжные системы); листовой металл и пластик	постоянный при потоке аксессуаров или проектный подряд на изделие; Польша, Чехия, Турция, Сербия — бюро aftermarket
Разработчик Python/OCCT	расширяет инструмент: правила, каталог разгрузок, стриминг, интеграции	средний; может вырасти из оператора	один, не сразу

Система сужает, что должен знать даже опытный конструктор: интерфейс платформы — данные, DFM — в валидаторе, цена — в контуре, библиотека даёт проверенные заготовки. Опыт тратится на концепцию и компоновку; опытный становится в разы быстрее, средний работает как опытный.

Что предлагать

Рабочее место без затрат с их стороны — стол через Guacamole, никаких лицензий и железа кроме ноутбука и проводного интернета (для страны, где SolidWorks стоит годовую зарплату — аргумент). Понятная система с правилами: продуктивен с первой недели. Рост по цифрам журнала, не по симпатии (раздел 8). Современный стек: OCCT, Python, Git, параметрика. Стабильная оплата в евро, долгосрочно, владение своим участком.

Как тестировать — самой системой, экзамен и рабочее место совпадают

Тест	Что проверяет
Реверс с ловушкой	STEP → реверс → объяснить вердикты; одна деталь — реально отклонённая порталом: найдёт ли причину и предложит ли правку, а не нажмёт кнопки
Грамотность по листу	чертёж с намеренными ошибками: отверстие угиба, нет разгрузки, нержавейка на алюминии без изолятора — сколько найдёт за 20 минут
Чтение Aufbaurichtlinien	кусок руководства → зоны и нагрузки в паспорт JSON с указанием страниц: аккуратность, языки, первоисточник
Честность	одна задача нерешаемая по условию: скажет «не сделать, потому что...» или нарисует что-то, чтобы сдать
Git-дисциплина	изменить параметр, закоммитить с внятным сообщением, объяснить дифф
Канал и пояс	замер задержки до сервера до разговора; выше 100 мс для роли с 3D — нет
Испытательный срок	месяц на реальных деталях; оценка по цифрам журнала — тем же, что определяют рост
Конструктор аксессуаров	оплаченная задача на неделю-две с вашим боксом: два-три варианта с компромиссами (масса, цена, время установки); выбранный проходит валидатор; посчитана цена; написан шаг инструкции; использованы библиотечные детали
Разработчик	одна задача: добавить правило валидатора с тестом (например, отступ отверстия от кромки) как pull request

36. Права на результаты и коммерческая тайна

Оговорка: не юридическая консультация; шаблон договора — у адвоката по немецкому и трансграничному праву. Каркас правильный вот такой. Половина этой системы уже построена — мы называли её защитой данных.

Юридический слой — один раз, при подключении

В Германии авторское право неотчуждаемо, но передаются исключительные права использования; изобретения — по отдельному режиму. Для внешнего подрядчика — рамочный договор, подписанный один раз: все результаты, созданные в системе, — с исключительными правами у RECADA; изобретения передаются и сообщаются; конфиденциальность; немецкое право, подсудность Аугсбург. Подпись квалифицированная электронная по eIDAS (Skribble или подобный) — работает через границу. Клик при входе как единственный договор — слабая позиция; клик как подтверждение уже подписанного — сильная.

Технический слой — подтверждение при каждом входе

Guacamole показывает условия перед сессией («работа в системе подпадает под рамочный договор от ...») и пишет в лог: кто, когда, с какого ключа принял. Доказательство, что человек знал условия в момент создания.

Происхождение результата — сильнее договора

Журнал проекта пишет автора и время каждой операции; Git — автора каждого коммита; запись сессий Guacamole — что происходило на экране. Три независимых лога: через год не нужно доказывать, кто спроектировал кронштейн. Работа физически не покидала сервер — создана в системе; спор «сделал дома, принёс потом» невозможен.

Коммерческая тайна — неожиданное следствие защиты данных

Немецкий закон о защите коммерческой тайны (GeschGehG) защищает секрет только при «надлежащих мерах по его охране». Пиксели наружу, личные ключи, запрет скачивания, запись сессий, договор при входе — это ровно те меры. Защита данных, построенная для другого, одновременно даёт правовую защиту библиотеки, паспортов и правил как коммерческой тайны. Без этих мер суд может сказать, что тайны не было.

Про «идею»

Идея сама по себе не охраняется нигде. Охраняется воплощение: чертёж и код — авторским правом, форма — промышленным образцом, механизм — патентом, остальное — как коммерческая тайна при тех самых мерах. «Подписывать идею» не получится; «всё, созданное в системе, принадлежит нам» — получится.

Порядок: рамочный договор с электронной подписью → подтверждение при каждом входе с логом → работа только в системе → результаты только в Git компании → оплата привязана к приёме. Для подрядчика в Турции или Польше адвокат добавит одну-две страницы про местное право. Доступ конструктора аксессуаров — стол с данными только его аксессуара и интерфейса, не вся библиотека; здесь Kasm уместнее Guacamole.

37. Монетизация: модели А, Б, В

Продолжение: Модель В

Продолжение: Модель В, четвёртый

38. Программа «Airholder Ready»

39. Поток вместо кодинга: измерить и уменьшить итерации

40. Управление проектом: доска по воротам вместо диаграммы Ганта

41. Датасеты CAD и CAD-агенты: что есть в мире и что из этого наше

42. Гибка листового металла в датасетах: BenDFM и KUL-bend

Второй обзор, посвящённый именно листовому металлу, добавил четыре источника, которых в разделе не было, и уточнил BenDFM. Ниже — каждый источник подробно: что внутри, как получить, что с ним делать у нас, какие ограничения. Это важная часть: она превращает «свой датасет» из идеи в план с конкретными входами.

BenDFM — уточнения по составу файлов

Файл	Содержимое	Что делаем
.stp	3D-модель согнутой детали	вход для probe / measure / fold2
_unfolded.stp	развёртка той же детали как STEP	эталон для нашего unfold: метрология «наша развёртка против их» по 20 000 деталям
_sequence.json	параметры исходной пластины и упорядоченная последовательность гибов с параметрами каждого	эталон для нашего FTG и порядка гибки из оригами-кинематики
_labels.json	коллизии с пуансоном и матрицей по каждому гибу, усилия гибки, коэффициенты сложности	проверка валидатора коллизий; калибровка «сложности» как компонента цены
предобработанные представления	графы для UV-Net (DGL) и облака точек для PointNeXt	сравнить наш обмер с тем, как деталь видят графовые сети — без обучения

Разбивка: BenDFM — 14 000 деталей с 2–8 гибоми, сбалансирован 50/50 по коллизиям инструмента; BenDFM-U — 6 000 деталей с 7–10 гибоми, сбалансирован по самопересечениям при развёртке. Инструмент в симуляции один: пуансон 10 мм с углом 90°, матрица 40 мм. Это значит, что метки коллизий верны для этой пары; для прессы Blehcon нужна своя таблица инструментов (уже предусмотрена в dfm_blehcon.json).

CADFS: человеческие последовательности листового металла по ключевым словам

Самая ценная мысль второго обзора. В 450 тысячах программ FeatureScript из Onshape есть класс, написанный специализированными инструментами Sheet Metal: sheetMetalStart, bend, flange, rip, unfold. Отфильтровать по тексту программ — и получают тысячи человеческих, не синтетических, последовательностей построения листовых деталей. Разница с BenDFM принципиальна: BenDFM показывает, как гнут случайно выбранные параметры; CADFS показывает, как гнут люди — какие радиусы выбирают, где ставят разгрузки и какого типа, в каком порядке идут фланцы, как решают углы. Лицензия открытая (Hugging Face, VladPyatov/CADFS).

Шаг	Действие	Результат
1	скачать CADFS, отфильтровать программы по наличию sheetMetalStart	подмножество листовых деталей — ожидаемо тысячи
2	разобрать вызовы bend / flange: радиус, угол, длина фланца, тип разгрузки	статистика реальной практики: распределение радиусов к толщине, углов, типов разгрузок
3	собрать геометрию из программ (FeatureScript выполняется в Onshape; альтернатива — взять готовые STEP/облака точек из датасета) и прогнать наш обмер	проверка распознавания на человеческих деталях; каталог разгрузок, снятый с практики
4	сопоставить со своей библиотекой	где наша школа совпадает с мировой практикой, где отличается — и почему

Ограничение: FeatureScript — язык Onshape, выполняется только там; для геометрии брать рендеры и облака точек из самого CADFS либо экспортировать STEP через Onshape для выборки. Разгрузки в Onshape Sheet Metal ставятся по встроенным правилам инструмента — это практика одного CAD, не всех; но именно её и стоит знать как ориентир.

ABC: фильтрация нашим же щупом

Обзор предлагает выделять листовые детали из миллиона моделей ABC по топологическому признаку: две параллельные эквидистантные поверхности (внутренняя и внешняя), соединённые торцевыми гранями фиксированной ширины, равной толщине, плюс цилиндрические переходы гибов между плоскими гранями. Это дословно наш виртуальный щуп из раздела 16: точка — нормаль $\times t$ попадает на поверхность → грань оболочечная; парование; двухцветная раскраска графа. То есть recada.ru уже является тем фильтром, о котором говорит обзор — без единой новой строки кода.

Что это даёт: прогон probe по подмножеству ABC (например, 50 тысяч моделей за ночь на EX44) выделяет реальные листовые детали из практики Onshape — не синтетику и не наши, а чужие, с полным разнообразием форм. Два применения: тест распознавания (сколько листовых деталей щуп находит, сколько пропускает, сколько ложных) и расширение каталога разгрузок и DFM-статистики за пределы нашей школы. Ограничение: STEP-файлы ABC под условиями Onshape Public Terms of Service — для внутреннего тестирования инструмента допустимо, для продукта и обучения нет.

Данные о пружинении: стартовые значения для rules/

Работы по прогнозированию пружинения при свободной гибке на прессе (V-die air bending): Mulidrán и соавторы, Li и соавторы — массивы конечно-элементных расчётов и натурных испытаний. В данных: геометрия заготовки, толщина, марка материала (высокопрочные стали HCT600X, алюминиевые сплавы, нержавейка), радиусы пуансона и матрицы, рабочий ход, конечный угол пружинения. Используются для обучения физико-информированных сетей (PINN, Theory-Guided DNN).

Что берём — не сети (PINN для одной цифры мы отклонили в разделе 4), а стартовые значения: таблица «толщина $\times$ радиус $\times$ материал $\rightarrow$ угол пружинения» для 1.4301 и 6063 в rules/dfm_blechcon.json. Записанное ранее правило «строка из практики Blechcon» остаётся, но теперь у неё есть литературный ноль: сначала цифры из измерений, потом калибровка по своим деталям. Ограничение честное: пружинение зависит от конкретной прессы, инструмента и партии металла; литература даёт порядок величины (для 1.4301 $t = 3$ при $R = 3$ — единицы градусов), не число для вашего цеха.

TRUMPF: извлечение детали из листа — новый класс правила валидатора

Промышленный датасет от TRUMPF, опубликованный в рамках исследовательского челленджа TU Dortmund: 15 293 технологические детали, 84 признака — геометрия внешнего и внутренних контуров (периметр, площадь, число вершин, моменты инерции), толщина, свойства сплава, параметры лазерной резки. Метка бинарная: удалось ли манипулятору автоматически вынуть деталь из листа после резки без заклинивания, микроперемычек и брака при выгрузке. Данные от производителя станков — редкость.

Для нас это класс проверки, о котором мы не думали: узкие контуры, мелкие детали, тонкие перемычки и неудачное соотношение площади к периметру влияют на выгрузку у поставщика — а значит на брак, время и цену. Blechcon режет на таких же станках. Действие: из 84 признаков и меток вывести простые пороги (минимальная площадь, минимальная ширина перемычки, отношение площади к периметру) и добавить в валидатор строку «риск заклинивания при выгрузке». Не обучать модель — извлечь пороги статистикой и проверить у Blechcon.

ESICUP и коллекции DXF: проверка оценки раскроя

Стандартизированные бенчмарки рабочей группы EURO по раскрою и упаковке: наборы Jakobs, Dighe, Albano, Dagli, Fu, Mao, Shapes — 2D-контур произвольной формы для нерегулярного нестинга с учётом зазора под рез. Плюс открытые коллекции DXF/SVG для лазера и плазмы (Deepnest, Sheet2Nest), очищенные от сплайнов. У нас предполётный нестинг уже предусмотрен (SheetNest, nest2d, CAD-N, u-nesting — раздел 2), а в разделе 33 записана эвристика «площадь заготовки к выпуклой оболочке». ESICUP — способ проверить эту эвристику на задачах с известным оптимумом: если наша оценка выхода материала

коррелирует с эталонным раскроем, ей можно верить в валидаторе. Полезно, не срочно.

Где обзор преувеличивает

«Analysis Situs / AAG Sheet Metal Benchmarks» как датасет — такого набора не существует. Analysis Situs — инструмент с функциями распознавания листовых элементов и построения дерева развёртки; мы отклонили его в разделе 4 как тяжёлую C++-платформу для того, что делают сто строк networkx. Функциональность есть, датасета нет. Запись здесь — чтобы не искать.

Сводный план действий по внешним источникам

Приоритет	Источник	Действие	Что закрывает
1	BenDFM (Zenodo)	регрессия unfold на 20 000 парах; обмер ~4 000 деталей с разгрузками	регрессия; каталог разгрузок; коллизии
2	CADFS, фильтр sheetMetalStart	статистика человеческой практики: радиусы, углы, типы разгрузок	DFM-правила из практики; каталог разгрузок с другой стороны
3	ABC, прогон щупом	50 тыс. моделей за ночь → реальные листовые детали	тест распознавания; разнообразие форм
4	пружинение (Mulidrán, Li)	стартовая таблица $t \times R \times$ материал → угол в rules/	ноль для калибровки по Blechcon
5	TRUMPF extraction	пороги по площади, ширине перемычки, площадь/периметр	новое правило валидатора: выгрузка
6	ESICUP, DXF-коллекции	проверка эвристики раскроя на задачах с известным оптимумом	доверие к оценке выхода материала
7	CC3D, AutoMate	из раздела 41: скан против STEP; сопряжения	входной контроль; граф связей

Все источники — полигон для тестов и статистика, не обучающие данные; каждый входит в library/external после проверки лицензии. Что остаётся уникальным в нашей библиотеке после всех семи: ответ портала, правки Blechcon, партия, поле. Внешние наборы делают инструмент проверенным на тысячах деталей; наши — делают его правильным для нашего производства.

Третий обзор — про производителей листогибного оборудования и инструмента: TRUMPF, AMADA, Bystronic, LVD, Salvagnini, WILA, Roller. Они почти не публикуют размеченных датасетов — коммерческая тайна. Но одна вещь доступна и закрывает конкретную дыру в нашем валидаторе: библиотеки геометрии инструмента. Ниже — все четыре группы источников подробно, с честной оценкой доступности и действием для нас.

Что у нас записано и чего не хватало

В разделе 2, в описании валидатора, стоит: «оболочечные модели инструментов запросить у Blechcon; каталоги Amada/Trumpf не нужны». В разделе 4 — «спуры матрицы как источник технологии: расстояние между линиями перехода выдаёт ширину V-паза». В разделе 42 — BenDFM с одним синтетическим инструментом: пуансон 10 мм, угол 90°, матрица 40 мм. Не хватало реальной геометрии инструмента поставщика, чтобы проверка коллизий из «оценки по синтетике» стала «проверкой на прессе Blechcon». Она есть в открытом доступе — у производителей инструмента.

Группа 1. Библиотеки инструмента WILA и Roller — главное

WILA (Нидерланды, инструмент под прессы TRUMPF и не только) выкладывает через официальный портал и приложение WILA Smart Tooling: 2D-контуры в DXF и 3D-модели в STEP пуансонов, матриц и адаптеров систем New Standard и American Standard; для каждого — геометрический профиль, рабочая высота, радиусы кромок, допустимый тоннаж на метр и минимальная длина полкигиба. Roller, ToolsPress, Precitools публикуют полные параметрические каталоги в STEP и DXF под кинематику четырёх стандартов: Trumpf-WILA, Amada-Promecam (европейский стандарт), Bystronic, LVD.

Что скачивается	Как используем	Что закрывает
STEP пуансона и матрицы	вставить в оригами-кинематику (раздел 4): деталь складывается пошагово, на каждом шаге пуансон и матрица ставятся по осигиба, коллизия — булева операция в OCCT	проверка «согнётся ли на этом прессе в этой последовательности» с реальным инструментом, а не с синтетикой BenDFM
Таблица «минимальная полка от ширины матрицы»	перенести в dfm_blechcon.json как правило с источником WILA	замена нашей оценки «15 мм при нержавеющей» на цифру производителя инструмента
Тоннаж на метр	правило: длинагиба × тоннаж на метр ≤ усилие прессы Blechcon	деталь, которую пресс физически не согнёт по длине, ловится до заказа
Радиусы кромок пуансона	связать наш R с реальной кромкой: $R \geq$ радиус пуансона при свободной гибке	радиусы из стандартной реихи инструмента, а не произвольные
Ширины V-пазов	таблица V = 6, 8, 10, 12, 16 ... мм; из раздела 4: спуры матрицы на чужой детали выдают V, теперь V сверяется с каталогом	реконструкция технологии чужих деталей проверяется по стандартной реихе

Единственный вопрос к Blechcon сводится к одному письму: «Какой стандарт инструмента у вас — Trumpf-WILA или Promecam — и какие ширины матриц и типы пуансонов в наличии?» После ответа таблица инструментов в dfm_blechcon.json заполняется из каталога WILA или Rolleri по названиям, без обмера и без запроса моделей у самого поставщика. Реалистично ожидать Trumpf-WILA: Blechcon в Золингене, TRUMPF в Дитцингене — типичная пара для немецкого цеха.

Группа 2. Базы таблиц гибки — недоступны напрямую, доступны через Blechcon

База	Что внутри	Доступ	Наш путь
TRUMPF Bendguide	более 5 000 верифицированных сочетаний марки стали, предела текучести, толщины и инструмента для расчёта пружинения и коррекции угла	внутри TruTops у владельцев станков; есть бесплатное приложение BendGuide для тоннажа и минимальных размеров	стартовые значения в rules/ из приложения; полная таблица — неявно в каждом ответе портала
LVD Easy-Form Laser / CADMAN-B	реальные замеры лазерными датчиками при адаптивной гибке: ход пуансона против фактического угла после снятия нагрузки, с учётом разнотолщинности проката и направления волокна	внутри прессов LVD	только если у поставщика LVD; тогда — просить экспорт по нашим деталям
Salvagnini MAC 3.0	динамические замеры усилий и пружинения на панелегибах P4/P2	внутри панелегибов	не наш класс оборудования

Вывод из группы 2: самые точные таблицы в мире живут внутри станков. У Blechcon они есть. Это ещё один аргумент за то, чтобы третий критерий работал как источник данных: каждая деталь, прошедшая через их портал и пресс, неявно содержит их таблицу коррекции. Калибровка K-фактора и пружинения по ответам портала (раздел 18) — единственный способ получить эти данные, не покупая станок.

Группа 3. САМ-пакеты с API — не покупать, просить экспорт

LVD CADMAN-B и CADMAN-FLOW (пакетный импорт STEP, автоматическое дерево развёртки, последовательность гибов, лог коллизий), Delem Profile-T3D (ЧПУ Delem: генерация последовательностей, экспорт развёрток, разметка), TRUMPF TruTops Bend (симуляция траекторий задних упоров и захватов роботизированных ячеек). Обзор называет это «стандартом генерации обучающих выборок в индустрии» — верно для тех, у кого есть лицензии, которые стоят как станок.

Наш путь: у Blechcon один из этих пакетов есть. Попросить экспортировать для наших деталей то, что он выдаёт — последовательность гибки, развёртку, лог коллизий. Это ровно те данные, что мы генерируем сами оригами-кинематикой и unfold. Сравнение двух источников — их САМ против нашего ядра — и есть калибровка: где совпало, ядро подтверждено; где разошлось, найдена либо наша ошибка, либо особенность их станка, которую надо записать в правило. Формулировка запроса: «Для каждой нашей детали — экспорт последовательности гибки и

развёртки из вашего CAM в любом формате; мы сверяем со своими расчётами, чтобы присылать вам файлы без итераций». Поставщику это выгодно.

Группа 4. Промышленные датасеты TRUMPF

TRUMPF Sheet Metal Extraction Prediction Dataset с TU Dortmund — уже в разделе 42: 15 293 детали, 84 признака, метка успешности выгрузки после резки. Доступен. «Исследовательские датасеты TRUMPF по гибке с SCCH и TÜV для PINN и валидации пружинения» — консорциумные данные европейских проектов по промышленному ИИ; публичных ссылок найти не удаётся. Не рассчитывать; если появятся — та же роль, что у литературных данных о пружинении: стартовые значения, не сети.

Как это встраивается в инструмент — конкретно

Шаг	Действие	Где в системе	Срок
1	письмо Blehcon: стандарт инструмента, ширины матриц, типы пуансонов, усилие пресса, длина стола	—	один день, ответ — неделя
2	скачать из WILA/Roller STEP и DXF по списку из ответа; сохранить в rules/tooling/	rules/tooling/*.step, tooling.json (имя, V, R кромки, высота, тоннаж/м)	вечер
3	перенести таблицы WILA в правила: минимальная полка от V, тоннаж на метр, стандартные R	dfm_blehcon.json	вечер
4	встроить в оригами-кинематику постановку пуансона и матрицы по оси каждогогиба и булеву проверку пересечения с уже согнутой частью	recada.py: fold2 + check_tooling	несколько дней
5	прогнать на своей библиотеке и на BenDFM с реальным инструментом вместо синтетического; сравнить метки	CI, регрессия	ночь
6	попросить у Blehcon экспорт из CAM по нашим деталям; сверить последовательность и коллизии	library/, паспорт детали: поле cam_export	по мере заказов

Результат: проверка коллизий из «оценки по синтетике» становится «проверкой на инструменте вашего поставщика». Валидатор получает три новых правила из первоисточника — минимальная полка, тоннаж, стандартные радиусы — и одну новую проверку — пересечение с пуансоном и матрицей по шагам гибки. Всё это до отправки на портал, то есть ещё один сдвиг проверок влево (раздел 39): доля деталей, проходящих с первого раза, растёт.

Что остаётся уникальным после всех трёх обзоров

Внешние источники теперь закрывают: геометрию инструмента (WILA, Roller), синтетическую разметку коллизий (BenDFM), человеческую практику гибки (CADFS), реальные листовые формы (ABC щупом), стартовые значения пружинения (литература, BendGuide), риск выгрузки (TRUMPF), проверку раскроя (ESICUP). Ни один из них не содержит: ответа портала конкретного поставщика на конкретную деталь, его правок, партии, полевого поведения. Библиотека RECADA остаётся единственной с этой меткой — и чем больше внешних полигонов проверяет инструмент, тем ценнее становится каждая такая метка, потому что она ложится на уже проверенное ядро.

Четвёртый обзор — самый широкий: от B-Rep-корпусов до конечно-элементных выборок и датчиков адаптивного ЧПУ. Здесь записаны все упомянутые в нём датасеты, включая те, что не наш процесс — чтобы не искать повторно и чтобы граница «наше / не наше» была зафиксирована. Одна настоящая находка — под наш материал: NUMISHEET 2025 с конститутивной моделью 1.4301.

Главная находка: NUMISHEET 2025 Industrial Benchmark — узел Mercedes-Benz T-Node из 1.4301

NUMISHEET — международная конференция по численному моделированию процессов листовой штамповки; её бенчмарки — эталон физической достоверности в мировой практике: строго синхронизированные экспериментальные данные оптических измерений, физико-механические паспорта металлов и CAD-модели штампов. Промышленный бенчмарк 2025 года посвящён силовому кузовному узлу Mercedes-Benz из аустенитной нержавеющей стали 1.4301 — нашей стали ног. Технологический цикл: глубокая вытяжка заготовки, трёхмерная лазерная обрезка контура, свободная упругая разгрузка. Открытый архив на Zenodo (DOI 10.5281/zenodo.17662411).

В архиве	Что это для нас
CAD-модели детали и инструмента: STEP, IGES, STL	референсная геометрия для проверки чтения и метрологии на сложной вытянутой форме
Параметры конститутивной модели 1.4301: кривая упрочнения, анизотропия, предел текучести	готовый паспорт материала для CalculiX — раздел 13 записывал калибровку валидатора по прочности как план; параметры 1.4301 брать больше не из справочника, а из верифицированного бенчмарка
Сканы реальной изготовленной детали после пружинения (оптические КИМ)	эталон для проверки, что наш расчёт пружинения в CalculiX совпадает с металлом — до того как верить ему на своих деталях

Технология там — вытяжка, не гибка на прессе. Но материальная карта от процесса не зависит: те же кривые идут в расчёт гiba, модальный анализ и прочность ног. Это единственный источник в четырёх обзорах, который даёт верифицированные данные именно по нашему материалу.

Бенчмарки пружинения NUMISHEET — эталоны для проверки алгоритмов

Бенчмарк	Содержание	Материалы	Нам
NUMISHEET 1993 U-Draw Bending	классический тест свободной U-образной гибки: геометрия штампов, режимы, профили после разгрузки; эталон для калибровки компенсации угла методами Displacement Adjustment и Spring Forward	низкоуглеродистая и высокопрочная сталь, алюминий	ДА: тест для расчёта пружинения в CalculiX, если будем считать, а не брать таблицей
NUMISHEET 2005 Benchmark 2	геометрия штампов и замеры пружинения поперечины днища	DP590, DP980, AA5182-O	косвенно: методика замеров
NUMISHEET 2008 Benchmark 2 (S-Rail)	скручивание и коробление S-образного лонжерона при усилии прижима 650–850 кН	автомобильные стали	нет: глубокая вытяжка сложной формы
NUMISHEET 2025 Scientific (DP800)	многомасштабные данные: EBSD, наноиндентирование, растяжение, сдвиг, послойное сжатие — для калибровки поверхностей текучести	двухфазная сталь DP800	нет: не наш материал; доступ по регистрации участников

Конечно-элементные корпуса — не наш процесс, зафиксировано

Датасет	Содержание	Почему не наше
ISMF TINY Dataset (Zenodo, 10,3 ГБ)	1 975 конечно-элементных расчётов инкрементального формообразования листа пуансоном по траектории с подвижной матрицей: поля перемещений, градиенты пластических деформаций, тензоры остаточных напряжений, профили пружинения; для обучения свёрточных и диффузионных операторов	инкрементальное формообразование — другая технология; поля напряжений с него на гибку на прессе не переносятся. Возможная роль: проверка настроек CalculiX на задаче с известным ответом

Датасет	Содержание	Почему не наше
Automotive Body-in-White Springback Dataset (NSF)	тысячи многостадийных нелинейных симуляций: формовка на геометрии NUMISHEET, позиционирование в кондукторах, прижим, контактная точечная сварка; параметризация по толщине, модулям, контактными силам	сварные кузовные под сборки; у нас болты и никакой сварки — осознанное решение раздела 31
RWTH Aachen MUC-Test	диаграммы цифровой корреляции изображений (DIC), лазерная экстензометрия, структуры RVE для стали DP800HNE; параметры анизотропного пластического течения	не наш материал; методика DIC может пригодиться при собственных испытаниях по ISO 11154

Уточнения к уже записанным источникам

WILA: каталог New Standard Premium, New Standard Pro и American Style; инструмент с угламигиба от остроугольных 28° до прямых 90°; паспорта нагрузки на метр; координаты стыковки для роботизированных захватов с системой Smart Tool Locator; доступ по регистрации на портале. ToolsPress и Rolleri — открытые порталы без регистрации, тысячи единиц; Amada-Promecam — европейский стандарт хвостовика с пазом 12 мм, Trumpf-WILA — быстродействующие гидравлические зажимы. TRUMPF TruBend: таблицы Bendguide калибруются датчиками ACB (Automatically Controlled Bending) с точностью $\pm 0,1^\circ$ на первом цикле; TRUMPF Extraction: цель — вероятность успешного захвата $\geq 0,8$ с оптимизацией полноты, потому что пропущенная заклинившая деталь останавливает всю гибкую ячейку. ESICUP: обучение с подкреплением и геометрические диффузионные сети с учётом kerf, термических поворотов и направления проката. Про «бенчмарки Analysis Situs с пограничной разметкой полок, гибов, подгибки кромок и рёбер жёсткости» — обзор снова описывает их как датасет; позиция прежняя: это функции инструмента, публичного набора нет.

Вывод обзора, который берём дословно: «first-part-right»

Индустрия отказывается от чисто геометрического машинного обучения в пользу гибридных физико-информированных подходов и идёт к слиянию трёх потоков: топологических B-Rep-графов деталей (BenDFM), конечно-элементных выборок (ISMF, NUMISHEET) и данных датчиков адаптивного ЧПУ (Easy-Form Laser, ACB, MAC 3.0) — ради концепции «first-part-right»: первое изделие без брака, без итераций на наладке. Это дословно наша метрика из раздела 39 — «доля деталей, проходящих портал с первого раза, цель девять из десяти». Мы измеряем то, что индустрия объявила целью, только на своём уровне: не станок, а поставщик. И наш гибрид тот же: детерминированное ядро, правила из практики, статистика поверх — раздел 15 и подтверждение из Лёвена в этом же разделе.

Полный реестр внешних источников после четырёх обзоров

Источник	Тип	Наш статус	Куда
BenDFM / BenDFM-U	B-Rep, синтетика, гибка	приоритет 1	регрессия unfold, каталог разгрузок, коллизии
CADFS(фильтр sheetMetalStart)	программы, человеческие	приоритет 2	DFM-статистика практики
ABC (прогон щупом)	B-Rep, реальные	приоритет 3	тест распознавания
WILA / Rolleri / ToolsPress	геометрия инструмента	приоритет 1 (после письма Blehcon)	коллизии на реальном инструменте, правила полки и тоннажа
NUMISHEET 2025 Industrial (1.4301)	материальная карта + сканы	приоритет 2	CalculiX: паспорт материала, проверка расчёта
NUMISHEET 1993 U-Bend	эталон пружинения	при расчёте пружинения	тест алгоритма компенсации
Пружинение (Mulidrán, Li), BendGuide	таблицы	стартовые значения	rules/dfm_blehcon.json

Источник	Тип	Наш статус	Куда
TRUMPF Extraction	табличный, промышленный	приоритет 3	правило «риск выгрузки»
ESICUP, Deepnest/Sheet2Nest	контуры раскроя	при проверке нестинга	доверие к оценке выхода
CC3D / CC3D-Ops	STEP + сканы	приоритет 2	входной контроль по скану
AutoMate	сопряжения	при модели соединений	граф связей
KUL-bend	реальный, закрытый	идея	время гибки как компонент цены
MM-CAD, CADCap-1M, SketchGraphs	мультимодальные	идеи	текст из паспорта, эскиз как запрос
ISMF TINY, BIW Springback, NUMISHEET DP800, RWTH MUC	FEM, другие процессы и материалы	не наше — зафиксировано	—
Fusion 360 (3), Text2CAD, DeepCAD и др. (раздел 41)	общий CAD	нет / лицензии	—
CAM-экспорт Blehcon (CADMAN, TruTops, Delem)	данные поставщика	просить с каждым заказом	калибровка ядра против их CAM

Итог четырёх обзоров в одной фразе: внешний мир даёт геометрию инструмента, синтетику коллизий, человеческую практику, материальную карту нашей стали и эталоны пружинения; не даёт — ответа конкретного поставщика на конкретную деталь. Всё первое проверяет ядро; только второе делает его правильным для RECADA.

Пятый обзор — «тотальный аудит» Zenodo, Mendeley, Figshare, GitHub, Hugging Face и архивов NUMISHEET, ESAFORM, IDDRG. Добавляет то, чего не хватало: точные DOI для скачивания, два китайских датасета распознавания элементов листового металла, бенчмарк ESAFORM и кластер по гибке труб. Ниже — только новое относительно предыдущих частей раздела, плюс исправление ссылок.

Новое по нашей теме — распознавание элементов листа

Датасет	Что содержит	Нам	Где
SMCAD (Northwestern Polytechnical University, Ma и Yang, 2024)	листовые CAD-модели в STEP с графами смежности граней; пограничная разметка: цилиндрические зоныгиба, вырезы, отбортовки, шарниры, рёбра жёсткости, проушины; для сегментации графовыми сетями в режиме Class-Incremental Learning	ДА: тест нашего распознавания на размеченных гранях — совпадает ли наш «гиб / фланец / разгрузка» с их метками; вход для каталога типов элементов	GitHub MaFangchang/SMCAD-dataset
HeteroMF (та же группа)	63 520 синтетических моделей с разнородными элементами листового формообразования: высечки, жалюзи, рёбра жёсткости, сложные гибы; стресс-тест классификаторов	ДА как стресс-тест: элементы, которых у нас нет (жалюзи, рёбра), проверяют, что шуп и AAG не ломаются на чужих формах	GitHub MaFangchang

Оговорка из самого BenDFM (записана в его статье): SMCAD строит разнообразие «добавлением элементов к небольшому фиксированному набору базовых геометрий», а не разнообразием фланцев и их взаимодействием — поэтому для производительности гибки он слабее BenDFM. Для распознавания элементов — наоборот, полезнее: у BenDFM только гибы и разгрузки, у SMCAD весь зоопарк.

DOI и точки скачивания — исправление и дополнение

Датасет	Точный адрес	Примечание
BenDFM / BenDFM-U	Zenodo DOI 10.5281/zenodo.18622958; репозиторий github.com/UGent-CVAMO/bendfm	в предыдущей части DOI не был записан — теперь есть
NUMISHEET 2025 Industrial (T-Node, 1.4301)	Zenodo DOI 10.5281/zenodo.17662411	подтверждено
ISMF TINY	Zenodo DOI 10.5281/zenodo.17917675; IT4Innovations, Чехия	не наш процесс, DOI для полноты
ESAFORM EXACT Benchmark	Zenodo DOI 10.5281/zenodo.6874577	см. ниже
Rotary Tube Bending (Университет Зигена)	Zenodo DOI 10.5281/zenodo.16614082; github.com/zeineddinoz/tubebend	трубы, не наше
Pressurized Tube Bending (Colorado Boulder)	Zenodo DOI 10.5281/zenodo.7136215	трубы, не наше
TRUMPF Extraction	портал челленджа TU Dortmund	прямая ссылка в обзорах не приведена; искать по названию челленджа
ESICUP	репозиторий рабочей группы EURO по раскрою и упаковке; серии Shapes, Albano, Blaz, Dighe, Fu, Jakobs, Swim, Dagli	добавлены серии Blaz и Swim
RL-Kirigami	github.com/milad1378yz/RL-Kirigami	лазерная микропрорезка; не наше

ESAFORM EXACT — материалы не наши, методика близкая

Экспериментальные данные и численное моделирование цилиндрической вытяжки алюминия AA6016-T4 и стали DC06: кривые упрочнения, профили фестонообразования, эволюция усилий. Материалы не наши (у нас 1.4301 и 6063), процесс — вытяжка. Ценность одна: это второй после NUMISHEET пример того, как выглядит открытый пакет «эксперимент + расчёт» для калибровки FEM — образец структуры для наших собственных испытаний по ISO 11154, когда они начнутся: что измерять, как публиковать паспорт материала, как сопоставлять расчёт с замером.

Гибка труб — зафиксировано как не наше

Rotary Tube Bending Dataset (Зиген): 318 натурных экспериментов ротационной вытяжной гибки труб на станке с ЧПУ, включая случаи брака; датчики усилий на дорне и прижимных колодках, траектории осей, 3D-геометрия сечения (овальность, утонение). Pressurized Tube Bending (Колорадо): 10 ГБ видео и диаграмм «момент — угол» для тонкостенных труб под внутренним давлением. У Airholder труб нет — платформа из профилей, ноги из листа. Если когда-либо появится трубчатый аксессуар (дуги, поручни), это стартовая точка; пока не наше.

Уточнения к инструменту

AMADA SheetWorks (Япония): библиотеки деформаций, выколоток, полок и матриц AMADA с контролем интерференции пуансона — интегрировано в среду Amada AP100/SheetWorks, отдельно не скачивается; актуально только если у Blechcon прессы Amada. WILA: обзор называет каталог открытым; предыдущий говорил о регистрации — при скачивании выяснится. Про «Analysis Situs Sheet Metal Feature Repository» — пятый обзор называет это «коллекцией тестовых B-Rep моделей для верификации алгоритмов распознавания»; тестовые модели в репозитории самого инструмента возможны, но публичного размеченного датасета под этим именем не находится. Позиция: проверить репозиторий Analysis Situs на наличие тестовых STEP, не рассчитывать на разметку.

Четыре кластера мировых данных — карта, в которой мы себя находим

Кластер	Что в нём	Наше место
1. Синтетические CAD/B-Rep корпуса	BenDFM, SMCAD, HeteroMF, CADFS-фильтр, ABC-щуп	полигон для распознавания, развёртки, коллизий; наша библиотека — тот же кластер, но с производственной меткой
2. Физические и FEM-бенчмарки	NUMISHEET, ESAFORM, ISMF TINY, BIW, RWTH	берём только карту 1.4301 и U-Bend как тест; остальное не наш процесс
3. Сенсорные данные реального оборудования	TRUMPF Extraction, KUL-bend, ACB, Easy-Form, MAC 3.0, трубогибы	доступно через Blechcon как экспорт CAM и через ответы портала; сами станков не имеем
4. Векторные библиотеки раскроя	ESICUP, Deepnest, Sheet2Nest	проверка оценки выхода материала

Ключевое наблюдение обзора, которое объясняет, почему наша библиотека имеет цену: «чертежи реальных серийных изделий защищены строгими NDA автоконцернов, оборонных предприятий и контрактных производств» — поэтому весь открытый мир состоит из синтетики, академических бенчмарков и обезличенных табличных данных. Реальная деталь с реальным ответом реального поставщика в открытом виде не существует нигде. У нас она будет с первого заказа.

Все ссылки, собранные из пяти обзоров и собственных поисков (сентябрь 2026). Формат: название — тип — адрес. Адреса приведены как в источниках; при скачивании проверять лицензию (правило раздела 41).

Название	Тип	Адрес
BenDFM	статья	arxiv.org/abs/2603.13102; Springer JIM doi 10.1007/s10845-026-02857-9
BenDFM	данные	doi 10.5281/zenodo.18622958; github.com/UGent-CVAMO/bendfm
KUL-bend / гибридный метод	статья	arxiv.org/abs/2605.12266
SMCAD	данные	github.com/MaFangchang/SMCAD-dataset
HeteroMF	данные	github.com/MaFangchang
NUMISHEET 2025 Industrial (T-Node)	данные	doi 10.5281/zenodo.17662411
NUMISHEET benchmarks (архив)	данные	архивы конференции NUMISHEET; 1993 U-Draw Bending, 2005 BM2, 2008 BM2 S-Rail
ESAFORM EXACT	данные	doi 10.5281/zenodo.6874577
ISMF TINY	данные	doi 10.5281/zenodo.17917675
Rotary Tube Bending (Зиген)	данные	doi 10.5281/zenodo.16614082; github.com/zeyneddinoz/tubebend
Pressurized Tube Bending (Colorado)	данные	doi 10.5281/zenodo.7136215
TRUMPF Extraction Prediction	данные	челлендж TU Dortmund (ссылку искать по названию)
RL-Kirigami	данные	github.com/milad1378yz/RL-Kirigami
ESICUP nesting benchmarks	данные	репозиторий EURO Special Interest Group on Cutting and Packing
Deepnest	инструмент/данные	deepnest.io; github
WILA Tooling / Smart Tooling	каталог инструмента	портал WILA
Rolleri, ToolsPress, Precitools	каталог инструмента	порталы производителей
MM-CAD	статья	Computer Graphics Forum, doi 10.1111/cgf.70523

Название	Тип	Адрес
CADSketchNet	данные	github.com/bharadwaj-manda/CADSketchNet
CAD-RAG	статья	wscg.zcu.cz/WSCG2025/papers/D11.pdf
ABC Dataset	данные	archive.nyu.edu/handle/2451/43778
A2Z-10M+	статья	arxiv.org/abs/2603.12605
Better STEP	статья	openreview.net/forum?id=RIKUK83N1L
DeepCAD	данные	github.com/ChrisWu1997/DeepCAD ; kaggle vitalygladyshev/deepcad
CADFS	данные	arxiv.org/abs/2605.01925 ; huggingface.co/datasets/VladPyatov/CADFS ; voyleg.github.io/cadfs
CADParser	данные	github.com/spicywagyu04/CADParser ; ijcai.org/proceedings/2023/200
Fusion 360 Gallery (Reconstruction, Segmentation, Assembly)	данные	github.com/AutodeskAILab/Fusion360GalleryDataset
Text2CAD	данные	arxiv.org/abs/2409.17106
Text2CAD-Bench	статья	arxiv.org/abs/2605.18430
CADCap-1M / DreamCAD	данные	huggingface.co/datasets/SadilKhan/CADCap-1M ; sadilkhan.github.io/dreamcad2026
Text-to-CadQuery	данные	arxiv.org/abs/2505.06507 ; github.com/Text-to-CadQuery
MFCAD / MFCAD++	данные	github.com/hducg/MFCAD
UV-Net	статья	arxiv.org/abs/2006.10211
AutoMate	данные	github.com/degravity/automate ; Zenodo
SketchGraphs / Vitruvion	данные	github SketchGraphs ; courses.cs.washington.edu (Vitruvion)
JoinABLE	статья	karliddwillis.com/joinable
ArchCAD-400K	данные	archiai-lab.github.io/ArchCAD.github.io ; arxiv.org/abs/2503.22346
CC3D / CC3D-Ops / CADOps-Net	данные	Uni Luxembourg CVI2 ; skazizali.com CADOps-Net
MCB / CADNet40v2	данные	см. Awesome-CAD
Awesome-CAD (список работ)	реестр	github.com/bertjiazheng/Awesome-CAD
BRep-GD	статья	computer.org/csdl/journal/tg/2026/07/1450536
GraphBRep	статья	academic.oup.com/jcde/article/13/1/259/8381229
Zero-to-CAD	статья	arxiv.org/abs/2604.24479
BenchCAD	статья	arxiv.org/abs/2605.10865
CadBench	статья	arxiv.org/abs/2605.10873
UniCAD	статья	arxiv.org/abs/2606.05058
CADENA (пошаговый реверс)	статья	arxiv.org/abs/2608.00799
Agent-Aided Design	статья	arxiv.org/abs/2604.15184
FillumaOne	статья	arxiv.org/abs/2606.17696

Название	Тип	Адрес
Graph-CAD	статья	arxiv.org/abs/2604.10075
cadrille	статья	arxiv.org/abs/2505.22914
mrCAD	статья	arxiv.org/abs/2504.20294
Обзоры: Geometric DL for CAD; AI-driven generation of 3D CAD; DL in 3D CAD Reconstruction	обзоры	researchgate; sciopen.com CVM.2025.9450521; mdpi.com/2076-3417/15/12/6681
The AI CAD Frontier	отраслевой обзор	spectrallabs.ai/research/The-AI-CAD-Frontier
OCCT 8.0.1	ядро	github.com/Open-Cascade-SAS/OCCT/releases
YouCanNotUnfold	библиотека	github.com/alexneufeld/YouCanNotUnfold
Onshape: Under the Hood	статья	onshape.com/en/blog/under-the-hood-how-collaboration-works
Zoo / KCL	продукт	zoo.dev
ОСР.wasm / Yet Another CAD Viewer; OpenCascade.js; Mayo; OCP CAD Viewer	просмотрщики	github (yeicor, ocjs.org, fougue/mayo, bernhard-42/vscode-ocp-cad-viewer)
Autodesk Platform Services	продукт	aps.autodesk.com
Aufbaurichtlinien VW / Ford BEMM / Stellantis	документы производителей	порталы для Aufbaushersteller (регистрация)
Front Runner Slimline 3 (Ranger P703)	конкурент	dometic.com
DX: исследование ИИ и производительности	исследование	DX, сентябрь 2026, 500+ команд

Правило работы со списком: каждый источник перед использованием проверяется на лицензию и актуальность; внешние наборы — полигон и статистика, не обучающие данные; ссылка в документе не означает, что источник уже скачан — статус скачивания ведётся в [library/external/README.md](#).

43. Риски и узкие места

Риск	Суть и защита
Обратный поток правок (№1)	Ручная правка в GUI не возвращается в params → два источника истины → смерть системы. Ручное — только песочница; в систему — через параметры/генератор; STEP на Blehcon — только из CI технически.
FreeCAD/SheetM etal	Community-аддон с пределами (SMBendWall, boolean OCC, 7-й фланец — предвестник: реальные крыши грязные, паспорт может расщепиться по комплектациям). Правило 80/20: экзотика руками вне конвейера — это нормально.
LLM-слой	Валидатор не ловит «корректную глупость» — человеческое ревью остаётся; время владельца — узкое место.
Bus factor = 1	Lead reviewer реально проводит приёмку без владельца минимум раз в месяц.
Scope creep	Система строится только вслед за реальным заказом; ворота со стоп-лоссами; чужие ревью концепта фильтровать вопросом «что убрать?» — оба присланных списка только добавляли.
Вы — IT-отдел	Gitea/CI/Kasm/VPN — скрытые 10-20% времени, учитывать в бюджете.
Заморозка интерфейса	Максимум внимания при минимуме информации; после продаж менять нельзя (legasy или слом совместимости).
Внешние зависимости	Shopify API, three.js, аддоны — версии фиксируются; восстановление из бэкапа тестируется ежеквартально.
Дешёвый аутсорс	Не бесплатен: ТЗ и ревью — часы владельца; смягчение — валидатор и сдельность.

Риски, добавленные по итогам практики (сентябрь 2026)

Инструмент — один файл без автоматических тестов; библиотека как регрессия — первое лекарство. Зависимость от версии OCCT: 7.8 в FreeCAD и 7.9 в ядре ведут себя по-разному — стоило полдня; фиксировать версию ядра и не смешивать. Третий критерий проверяется руками через чужой портал, который может измениться без предупреждения — вести журнал ответов портала как данные, чтобы изменения были видны.

44. Дорожная карта

Ближайший порядок работ — конкретно

Шаг	Что	Зачем
1	установка сервера: ОС, ключи, ufw, WireGuard, restic на Storage Box (Хельсинки)	база и бэкап в другом ДЦ
2	разделить ядро и интерфейс, очередь задач, постоянное хранилище	многопользовательность
3	сканер библиотеки: пакетный обмер, таблица, статистика	данные вместо догадок
4	регрессия на CI по библиотеке: процент прохождения трёх критериев	не ломать работающее
5	калибровка K-фактора по ответам портала	развёртки точнее
6	каталог разгрузок → генерация зоныгиба по правилу	параметрика углов проходит производство
7	гибридный workbench FreeCAD (вызов внешнего инструмента)	инженер рисует руками, считает ядро
8	паспорта автомобилей в библиотеке → редактор изделия на данных	багажник как параметры
Этап	Содержание	Статус
0. Фундамент	Реверс Bracket_L/R (метрология 100%), wind_side_v2, Skeleton_Ranger + MasterParams, паспорт Ranger, DFM-правила как skills; решения по железу зафиксированы (EX44-критерий)	Сделано
1. Хребет	Локальный старт на ThinkPad: git init + структура + build123d + первый генератор L-кронштейна с автосверкой (вечер №1); Gitea+WireGuard (Strato или сразу EX44 по прибытии); первый CI «params → STEP+GLB»; step_probe.py (обмерщик STEP на AAG) как первый модуль библиотеки; ziprey; 7-й фланец (BSpline)	Первый шаг
2. Валидатор и каталог	Валидатор DFM+ (cad-khana + зазор 20/упаковка GLS (40 кг/коробка)/креплёж) + геометрический дифф в PR; spec.yaml + автогенерация приёмочного прогона из constraints; catalog.json; статусы двух осей + webhook Shopify; SheetNest; рендер-конвейер; автосхема конвейера (Mermaid); архив счетов Blechcon и тарифа GLS (пороги надбавок — в rules/, честная доставка в CM1); параллельный трек испытаний: собрать существующие отчёты (статика, модальный) как эталоны калибровки; запланировать физические испытания ISO 11154 с расчётным обоснованием	
3. Рабочие места	Kasm CE на EX44 (пентест лично, вкл. Python-консоль); egress-файрвол; RECADA-workbench; Claude в VDI-образе с лимитированным ключом	
4. Люди	Onboarding-пакет EN; воронка Upwork+hh.uz с пробой канала; автотест с автоприёмкой; GDPR-оформление; такт 2 недели; YAML-инструкции; светофор нагрузки; скрипты CI: складской webhook и онбординг	

Этап	Содержание	Статус
5. Масштаб	Вторая машина по конвейеру; lead reviewer; библиотека узлов → нодовый composep (CPQ с PDF-КП для дилеров; быстрый внутренний прототип панели параметров — Streamlit, выход строго в params.json/Git, чтобы не плодить второй источник истины); FEM в валидаторе (CalculiX+Gmsh: статика с нормативной аэронагрузкой, Мизес с запасом 1.5, прогиб подтверждает зазор 20 мм, модальная проверка $f_1 \geq$ порога по Струхалю, автоотклонение PR с картой напряжений; калибровка по существующим отчётам); ИИ-агент паспортов (STL → распознавание отверстий/нормалей → черновик vehicle_passport + PR; подпись человека обязательна); второй поставщик металла; публикация спецификации интерфейса; сообщество; узел Сингапур/Мумбаи при 40+ ч/мес; RISE-L/i9 при 10+ сессиях; контент-маркетинг; ISO-формализация по запросу рынка	

Формула проекта: мозг и знания — в Германии (структурно самой дешёвой серверной стране), руки — где угодно (пиксели вместо файлов, любой CAD на входе, STEP на выходе), приёмка — автоматическая, рост людей — по логам, а не по резюме. Деталь описывается один раз параметрами; производство, конфигуратор, инструкции и склад — производные одного конвейера. Владелец управляет трубой: приоритеты на входе, критерии приёмки, кнопка релиза.

RECADA / Airholder · Полное описание проекта v25 · Сводка обсуждений, август 2026 · Ставки, цены, пинги — ориентировочные на дату; цены Hetzner/OVN проверять при заказе