

2.3.1

Прогноз сроков проекта

Technical — Projects

Технические заметки по модулю проектов (синхронизация с ask/documentation/technical/, август 2026).

User-facing руководство

Руководство пользователя: 3.5. Управление проектами.

Контуры

AI estimate LLM design (v2) AI estimate progress checklist Унифицированная карточка задачи и вехи Требования, результаты и CR Прогноз сроков (Schedule) Health и портфель

Преднастройки (пресеты)

Код: projects/presets.py. Поле проекта: ProjectConfig.management_preset (lite / standard / pro / expert). Глобальный дефолт: ProjectSettings.PROJECT_MANAGEMENT_PRESET (по умолчанию standard). Выбор в /settings/projects/ синхронизирует связанные feature flags (GLOBAL_FLAGS_BY_PRESET). Новые проекты получают глобальный пресет; в визарде и настройках проекта — picker из 4 карточек (apply_preset_to_config меняет только RM/FM/согласования). Навигация и вкладки гейтятся через CAPABILITIES_BY_PRESET / project_has(). Пользовательская документация: Преднастройки функционала.

Кратко по флагам (код projects/feature_flags.py)

Флаг	Default	Назначение
PROJECT_MANAGEMENT_PRESET	standard	Глобальный пресет модуля (см. выше)
PROJECT_FEATURE_SCHEDULE	True*	Прогноз сроков / Гант forecast
PROJECT_FEATURE_REQUIREMENTS	True*	Требования и deliverables

PROJECT_FEATURE_HEALTH_UI True* Обзор здоровья / портфель
PROJECT_FEATURE_ECONOMY_SHELL True* Shell экономики
PROJECT_RESOURCE_MANAGEMENT True* Глобальный ресурсный контур (+ per-project override)
PROJECT_AI_ENABLED False* ИИ (оценка, табель, explain...)
PROJECT_KB_ASSIST_ENABLED True* Знания проекта
TIMESHEET_SUGGEST_V2 True (если AI on) Подсказки табеля
* Фактические значения после смены глобального пресета переписываются матрицей GLOBAL_FLAGS_BY_PRESET (для standard ресурсы и AI выключены, требования выключены; для pro/expert — включаются по матрице).

Миграции (ориентир август 2026)

Среди прочего: nullable manager, resource_management_enabled / finance_management_enabled, status report + templates + project documents, portfolio status report, search vector, timesheet suggestion models, schedule forecast v2 fields, executor_from_team_only, AI estimate feedback, task embeddings, management_preset + seed PROJECT_MANAGEMENT_PRESET / PROJECT_KB_ASSIST_ENABLED.

Project Health и портфель

Project Health и портфель

Дата актуализации: август 2026.

Назначение

Агрегированный read model здоровья проекта из детерминированных schedule / economy / coverage сервисов. Портфель проектного офиса работает по снимкам без пересчёта в запросе.

Пакет

projects/health/:
generate_project_health_snapshot portfolio rows / KPI published snapshot lookup

UI

Canonical landing: /projects//overview/ Portfolio: /projects/portfolio/ Sidebar: Portfolio health, My timesheet

Права

Независимые object permissions: finance / schedule / requirements / health view.

Feature flags

PROJECT_FEATURE_HEALTH_UI PROJECT_FEATURE_ECONOMY_SHELL
PROJECT_AI_ENABLED (default False; preview-only)

Прогноз сроков проекта

Прогноз сроков проекта (Schedule)

Дата актуализации: август 2026.

Назначение

Детерминированный контур прогноза сроков поверх WBS, ETC, календаря и зависимостей задач. AI не рассчитывает даты — только объясняет уже посчитанные причины.

Пакет

projects/schedule/
calendar.py — рабочие дни / Holiday capacity.py — доступность назначенных ресурсов forecast.py — leaf forecast + агрегация graph.py — зависимости Task/TaskDependency critical_path.py — critical path / float snapshots.py — versioned ProjectScheduleForecastSnapshot summary.py — read model для UI/API invalidation.py — transaction.on_commit + Celery

Правила

Official ETC: manual → AI accepted → calculated Нет ресурса → unknown forecast, без фиктивной даты Schedule baseline (Baseline / BaselineTask) immutable относительно current Task.start_date/duedate Активация baseline не перезаписывает текущие даты задач (Baseline.update_task_dates — no-op)

UI / API

Вкладка «Сроки / прогноз» в плане проекта: `/projects//planning/?tab=schedule`
URL `/projects//schedule/` редиректит на эту вкладку; HTMX-фильтр таблицы прогноза обслуживается по старому URL `Gantt payload` включает `baselineStart/End, forecastStart/End, isCritical` REST: `schedule summary / forecast / history` через `project viewset`

Feature flag

`PROJECT_FEATURE_SCHEDULE` (default True).

Прогресс: оценка трудозатрат AI №1

Прогресс: оценка трудозатрат AI №1 (v2)

Дизайн: `ai-estimate-llm-design.md` Модель: v2 — история → P50/P80 → LLM features (не «LLM пишет часы») ТЗ: §45–54, §70–74, §94–95, §101, §110, §112
Обновлено: 2026-08-25 Легенда: [x] сделано · [~] частично · [] не сделано · [!] вред / ложь UI

Сводка

Область Статус

Фазы 1–15 (дизайн) ☐

Task embeddings (hybrid Jaccard+vector) ☐

Bias calibration из feedback ☐

ask-doc sync ☐

Контур AI №1 v2 закрыт. Дальше — эксплуатация: backfill embeddings, накопление feedback, тюнинг порогов. Флаг LLM-оценки: `PROJECT_AI_ESTIMATE_LLM` (default True; в тестах False).

Команды эксплуатации

```
# leave-one-out backtest python manage.py ai_estimate_backtest --limit 20 --csv
# embeddings для задач с утверждённым фактом python manage.py
backfill_task_embeddings --limit 500
# sync actuals + bias factors python manage.py ai_estimate_calibration --sync
Celery beat:
projects-task-embedding-backfill-weekly (вс 05:30)
projects-ai-estimate-feedback-actuals-daily (ежедневно 04:20)
Сигнал: Task post_save → update_task_embedding_task (lifecycle leaf). Миграции:
0053_project_ai_estimate_feedback, 0054_task_embedding.
```

Документация

ask:	documentation/technical/ai-estimate-llm-*.md	ask-doc	user:
	3-5-Upravlenie-proektami/ii-ocenka-trudozatrata.md	ask-doc	tech:
	technical/projects/ai-estimate-llm-*.md		

G. Журнал

Дата Изменение
2026-08-25 Фазы 1-15
2026-08-25 Task embeddings hybrid + bias calibration + ask-doc
2026-08-25 Сигнал TaskEmbedding + beat sync feedback/actuals + drawer tests

Требования, результаты и change requests

Требования, результаты и change requests

Дата актуализации: август 2026.

Назначение

Реестр требований проекта с coverage, критерии приёмки, deliverables, readiness, closure gate и change requests с impact preview.

Пакет

projects/requirements/
versioning / approve / link / coverage / readiness / closure gate deliverables.py —
accept deliverable, CR impact preview/approve, closure override

Правила

Mandatory critical без implements-связи = uncovered Завершённая задача не принимает требование без passed criterion Approval CR обновляет Current Scope/Plan и может создать budget revision; baseline не меняется Closure override требует reason + project.closure_override

UI

/projects//requirements/ — матрица: создание, approve, связь с lifecycle-задачей, переход к созданию результата /projects//deliverables/ — результаты (отдельная сущность), создание с опциональной связью к требованию, CR, closure gate

Approve требования не создаёт deliverable и не покрывает требование: покрытие = связь implements с задачей; результат сдачи заводится отдельно.

Feature flag

PROJECT_FEATURE_REQUIREMENTS (default True).

Унифицированная карточка задачи и вехи

Унифицированная карточка задачи и вехи

Дата актуализации: август 2026. Источник: UX/UI карточка задачи/вехи v3.

Канон

URL:	/projects//tasks//control/	(projects:project_task_control)	Backend:
	projects/task_control.py	→ build_task_control_card	Шаблон:
	templates/projects/economy/task_control.html		+
	fragments/task_control_overview.html	API:	GET
	/api/projects/{id}/tasks/{task_id}/control-card/		

Одна карточка для leaf / summary / milestone. Различие — видимость блоков, не отдельный экран.

Статус

Единственный вычисляемый статус (без «операционного состояния»):

Код Подпись Условие

cancelled Отменена task.canceled

done Выполнена task.completed

rework Возвращена на доработку task.backward > 0 (открытая)

in_progress В работе есть start_date / факт часов / progress > 0

not_started Не начата иначе

Блокировка зависимостью — отдельный бейдж «Заблокирована», не статус.

Права: экономика

Блок «Экономика задачи» (себестоимость / продажа / маржа) видят только:

руководитель проекта (project.manager); проектный офис (группа Projects-Edit); Rules-TopMan; superuser.

Исполнитель не видит финансы даже при наличии project.fin_view. Макет блока: строки «подпись слева / сумма справа», группы через разделитель; прогноз (ЕАС) жирным; отклонение отдельной цветной строкой под прогнозом (для маржи — Р и п.п.).

Участники

Правая колонка карточки задачи/вехи — блок Участники (не в шапке со статусом):

Текущий исполнитель (для вехи — «Ответственный за веху», только если назначен) Контролёр — только если назначен, без «не назначен» Постановщик — author, иначе created_by

Один человек с несколькими ролями показывается один раз с подписями через «·». Если текущий пользователь — контролёр, бейдж «Вы контролируете эту задачу». Для поручения те же роли выводятся компактно в шапке карточки (исполнитель → постановщик → контролёр). В списке поручений есть колонка постановщика и фильтры «Мне поручено / Я поручил / Я контролирую».

Веха

Тот же каркас; скрыты быстрый ввод времени и прямой Plan/Fact ввод. Остаются сроки, health, связи, влияние на проект.

«Что дальше»

Вычисляемый блок (не поле модели). Показывается только при уверенном сигнале: активный blocker, не начата у срока, ETC > плана. Иначе скрыт.

Редактирование lifecycle

Единая модалка для задачи/вехи ЖЦ (и с вкладки «План», и из меню control-карточки):

GET /tasks//edit-data/?return_to=planning|control POST /tasks//save-edit/ JS: static/js/lifecycle_task_edit.js Шаблон: templates/projects/planning/modals/task_edit_modal_content.html

В форме: название, описание, исполнитель, контролёр, даты/длительность, тип результата (веха). Не в форме (другие процессы/контролы): статус completed/canceled, приоритет, прогресс, вложения, author/project/source, перенос срока исполнителем. Popup tasks:task_update (TaskForm) остаётся для не-lifecycle задач (SD/meeting и т.п.).

Обсуждение (комментарии)

На вкладке «Обзор» — shared composer как у заявки/карточки задачи:

ответ (reply) и дерево веток; @-упоминания сотрудников/ролей задачи; быстрое форматирование; вложения (draft → TaskAttachment); важные комментарии.
POST: projects:project_task_control_comment (HTMX → фрагмент ленты).
Лента/reply/attach: URL-ы приложения tasks (comments_feed, comment_reply, ...).

Целевой дизайн: оценка трудозатрат проекта

Целевой дизайн: оценка трудозатрат проекта (AI №1) — v2

Статус: design Версия модели: v2 (reference-class + LLM features, не «LLM пишет часы») Дата: 2026-08-25 Прогресс: ai-estimate-llm-progress.md ТЗ: «Переработка планирования и учёта ресурсов проектов» (§45-54, §70-74, §94-95, §101, §110-112) Связанные: planirovanie-i-uchet-resursov-proekta.md, ask-doc navigaciya-proekta-2026.md

1. Принцип

LLM понимает работу → история компании даёт норму → статистика даёт диапазон → человек принимает решение.

Слой Ответственность

LLM Тип работы, size, quantity, complexity, novelty, roles, gaps

История (actual hours) Норма: task analogs, unit rates, work_type × size

Статистика P20 / P50 / P80, evidence confidence

Человек Акцепт / правка / уточнение параметров

Не делать: есть task → 8 часов. Константа 8 ч создаёт ложную точность и должна исчезнуть из оценочной логики. Не менять: архитектуру job → preview → accept → draft (ProjectAIEstimateJob, HTMX wizard, права, флаг). Меняются estimate_project() и контракт preview/Accept.

2. Контракты ТЗ (обязательные)

§ Требование Как в v2

45 propose → confirm → save без изменений

46 источники: описание, WBS, документы, история history = основной источник часов; LLM — понимание score

47 кнопка в плане/бюджете Экономика hub есть; добавить toolbar Ресурсного плана

48 3-step wizard каркас сохранить; step 3 — новый UI (P50/P80, evidence)

49 confidence + range evidence confidence + P50/P80 (не $8 \times 0.85 \dots 1.2$)

50 выбор строк → draft lines Accept пишет в task_id строки

51-54 аналоги двухуровневые: task analogs (часы) + project analogs (sanity)

70-72 service layer, schema LLM output = features, не hours

73 LLM не считает деньги часы/роли → rate engine после Accept; cost_preview детерминированно

74 аудит + model/prompt_version; feedback estimate→human→actual
94-95 async + fallback Celery; без истории — «недостаточно данных», не 8ч
101 нет auto-lines; cost deterministic сохранить

3. As-is (проблемы)

Проблема Почему критично

8 ч на любую leaf-задачу Ложная точность; «концепция» = «подпись в документе»

Лимит [:12] Оценка не покрывает score проекта

Documents checkbox + «Изучаю документы...» UI врёт — текст не читается

Асепт → все строки на первую lifecycle-задачу Preview и план означают разное

Один ресурс Labor (AI estimate) Теряется role mix

Similarity rerank по BAC hours Круговая зависимость для первичной оценки

Confidence 0.5 / 0.55 Не evidence-based

Аналоги только project-level Не используются approved actual hours похожих задач

4. Целевой результат preview

4.1. Уровень проекта

Оценка проекта P50 — 486 ч P80 — 635 ч Reference class (похожие проекты) P50 — 510 ч Evidence confidence — 74/100

Покрывание: 82% — задачи с хорошими аналогами 11% — слабые аналоги 7% — без исторических данных

⚠ Bottom-up P50 (440) существенно ниже reference class (750). Возможная причина: неполная WBS или неучтённые работы.

Важно: evidence confidence — качество доказательной базы, не «вероятность уложиться в P50». Вероятностный смысл несут P50 / P80. Нельзя просто суммировать P80 всех задач → чрезмерно консервативный проект. v1: project_p50 ≈ sum(task_p50); project P80 — из reference class и/или упрощённой агрегации. v2+: Monte Carlo по распределениям задач (Celery).

4.2. Уровень строки (работа × роль)

Работа Роль P50 Диапазон (P20-P80) Достоверность Основание

Интеграция Аналитик 12 8-18 78 8 похожих задач

Интеграция Разработчик 38 24-61 72 8 похожих задач

Интеграция QA 10 6-16 70 5 похожих задач

Асепт создаёт отдельные plan lines: task_id + role + hours.

4.3. Drawer строки («Почему N часов?»)

список task-аналогов (факт, similarity); факторы сложности / новизны; missing information; что повысит точность (уточняющие вопросы); правка часов + опциональная причина (scope / нерелевантная история / эксперт / другое).

5. Pipeline estimate_project() v2

estimate_project() | └─ collect_scope() | └─ ALL leaf lifecycle tasks (без [:12]) | └─ project description / charter | └─ relevant document chunks (retrieval по задаче) | └─ understand_scope() # LLM, optional если AI off → heuristics | └─ structured task features (не hours) | └─ retrieve_task_analogs() # embeddings + metadata filters | └─ estimate_tasks() # детерминированный слой | └─ 1. quantity × unit rate | └─ 2. distribution actual hours аналогов (≥N) | └─ 3. work_type × size norms | └─ 4. top-down от project analogs (низкая confidence) | └─ 5. LLM 3-point / экспертный hint + LOW confidence | └─ 6. «Недостаточно данных» (не 8ч) | └─ estimate_roles() # role mix на строку/пакет | └─ reference_class_project() # sanity vs completed peers | └─ calibrate_uncertainty() # P20/P50/P80 + evidence confidence | └─ simulate_project() # later: Monte Carlo → project P50/P80 | └─ build_preview() | └─ rows (task_id, role, hours, ranges, evidence) | └─ project totals | └─ reference-class warning | └─ assumptions / gaps / clarifying questions | └─ audit
Job / poll / Accept-endpoint остаются; меняется наполнение job.result.

6. Двухуровневые аналоги

6.1. Task-level (основной источник часов)

Для каждой leaf-задачи (или estimate-package): «Разработка API интеграции с ERP» → исторические задачи с approved actual hours → распределение фактов → P20 / P50 / P80 → типичный role mix

Leave-one-project-out: текущий проект не аналог сам себе.

6.2. Project-level (контроль здравого смысла)

Bottom-up Σ P50 vs reference class завершённых проектов:

не автозамена часов; сигнал РП: «структура работ существенно ниже/выше истории».

6.3. Similarity для initial estimate

Опирается на: тип, scope, deliverables, WBS, отрасль, технологии, complexity. Не давать бонус за близость BAC hours при первичной оценке (круговая зависимость). BAC-similarity — только в режимах, где BAC уже известен независимо (прогноз исполнения).

7. LLM: features, не hours

Пример structured output (валидация schema §72): { "task_id": 123, "work_type": "external_system_integration", "deliverable": "working API integration", "size": "L", "quantity": 3, "quantity_unit": "integration_methods", "complexity": 4, "novelty": 3, "external_dependencies": 2, "required_roles": [{"role": "analyst"}, {"role": "developer"}, {"role": "qa"}], "risk_factors": ["API docs incomplete"], "missing_information": ["number of exchanged entities"] }

Батчи по 10–15 задач в LLM; aggregation по всем leaf. Крупные summary без детей → предложение estimate packages (декомпозиция) без автозаписи в WBS; пользователь может «Добавить задачи в ЖЦ» или оставить только структуру оценки.

8. Иерархия fallback (без 8 часов)

1. quantity × исторический hours/unit 2. $\geq N$ хороших task-аналогов → квантили actual 3. статистика work_type + size (XS...XL) 4. top-down от похожих проектов + low confidence 5. LLM 3-point suggestion + явно LOW confidence 6. «Недостаточно данных для оценки»
Шаг 6 лучше ложного 8 ч / medium.

9. Evidence confidence (расчёт, не константа)

Компоненты (пример 0–100):

Фактор Смысл

Similarity близость аналогов

Sample size число качественных аналогов

Dispersion разброс фактов (P80/P20)

Input completeness полнота описания / документов / quantity

Низкая достоверность → подсветка + «уточните N параметров» (clarifying questions). Ответы пользователя сужают диапазон (пересчёт estimate без полного перезапуска job — целевой UX).

10. Quantity × unit rate

Где удаётся извлечь естественную единицу (лист, endpoint, сценарий, требование...): quantity × historical P50 hours/unit → P50 + квантили unit-rate → P20/P80

Использовать существующий planning_mode=qty, где уместно. Без надёжного unit-rate — не форсировать; fallback на task analogs (иначе ложная точность). Нормы work_type × size (XS–XL) накапливаются из approved actual и со временем адаптируются.

11. Документы

До retrieval: скрыть checkbox documents или убрать текст «Изучаю документы...». Целевое: task context → search по документам проекта → 3–10

релевантных фрагментов → LLM extraction → scope / quantities / constraints
Не скармливать весь комплект документов целиком.

12. Асепт (критичный контракт)

Каждая proposed row: { "task_id": 123, "task_name": "...", "role_id": 12, "role_hint": "developer", "estimated_hours": 34, "p50": 34, "p80": 52, "evidence_confidence": 78, "method": "task_analogs", "selected": true }

Асепт:

пишет в task_id строки (не first leaf); создаёт линии по ролям (каталог / role_hint), не одну Labor (AI estimate) если роль известна; даты: task.start/finish → project dates → только потом fallback; убрать today + 30 + idx как основной период; DRAFT only; ACTIVE baseline не трогать.

Разделение:

Estimate — сколько каких компетенций; Allocation (отдельный AI) — кто именно.

13. Feedback loop и backtest

После закрытия работ сохранять: AI P50 / AI P80 / Human accepted / Actual approved

Метрики: AI error, human adjustment, P50/P80 coverage, bias. Перед продакшен-включением новой модели — backtest на завершённых проектах (состояние на момент планирования, без будущего факта), leave-one-project-out. Сравнить: 8h/task vs median type vs task analogs vs analogs+LLM features vs human.

14. Структура кода

projects/ai/ __init__.py # ProjectAIService gateway estimate/ scope.py # collect_scope, leaf enumeration features.py # LLM / heuristic understand_scope task_analogs.py # retrieve + distributions unit_rates.py # quantity × hours/unit norms.py # work_type × size tables estimator.py # fallback hierarchy → row estimates project_reference.py # reference-class sanity uncertainty.py # P20/P50/P80, evidence confidence simulate.py # Monte Carlo (фаза позже) preview.py # build_preview JSON schemas.py # Pydantic features + preview rows prompts.py context.py # document retrieval helpers (или внутри estimate/) run_ai_estimate_job вызывает ProjectAIService.estimate_project как сейчас.

15. Контракт job.result (preview JSON)

```
{ "project": { "p50_hours": float, "p80_hours": float, "p20_hours": float | None,
"evidence_confidence": int, # 0-100 "reference_class_p50": float | None,
"reference_class_warning": str | None, "coverage": {"strong_pct": float, "weak_pct": float, "none_pct": float}, }, "rows": [{ "task_id": int | None, "task_name": str,
"role_id": int | None, "role_hint": str | None, "resource_type": str, "estimated_hours": float, # = p50 для Accept "p20": float | None, "p50": float, "p80": float,
"evidence_confidence": int, "confidence_label": "high|medium|low", "method": "unit_rate|task_analogs|type_size|project_topdown|llm_hint|insufficient", "evidence":
```

```
{ "analog_count": int, "analog": [{"task_id", "project_id", "actual_hours",
"similarity"}], "reasoning": [str], "missing_information": [str],
"clarifying_questions": [str], "selected": True, "is_estimate_package": bool, #
декомпозиция без WBS }, "similar_projects": [...], "gaps": [str], "assumptions":
[str], "notes": str, "sources": {...}, "llm_used": bool, "model_version": str,
"wizard_step": 3, "cost_preview": {...} | None, # deterministic rate engine "audit":
{...}, }
```

Обратная совместимость UI: estimated_hours = P50; старые поля range_low/high можно маппить из P20/P80.

16. UI step 3 (целевой)

Итого P50 / вероятный диапазон / достоверность / reference class. Таблица работ с основанием и подсветкой low confidence. Drawer: evidence + clarifying questions + правка + причина. «Принять выбранное» → Ресурсный план (draft). Кнопка: Экономика hub + toolbar «План / Бюджет» (§47).

17. Фазы внедрения (по отдаче)

Фаза Содержание Нужен LLM?

- 1 Accept fix task_id→task_id, даты из задачи, role если есть Нет
- 2 Убрать 8h + [:12] all leaf; insufficient вместо 8 Нет
- 3 Task analogs embeddings задач + actual → P50/P80 Нет
- 4 Evidence confidence similarity/N/dispersion/completeness Нет
- 5 Documents honesty скрыть/переименовать до retrieval Нет
- 6 LLM features classify size/type/quantity/roles/gaps Да
- 7 Documents retrieval chunks → extraction Да
- 8 Role mix в Accept несколько линий на задачу Частично
- 9 Project reference-class sanity warning Нет
- 10 Feedback + backtest estimate/human/actual; leave-one-out Нет
- 11 Unit rates + XS-XL norms quantity×rate, таблицы норм LLM для quantity
- 12 Clarifying questions UX сужение диапазона Да
- 13 Monte Carlo + calibration project P80 из симуляций Нет
- 14 UI step 3 redesign drawer, coverage, reference banner —
- 15 Кнопка в Ресурсном плане §47 —

Даже до Ollama фазы 1-5 + 9-10 дают систему сильнее текущего stub.

18. Тесты

Accept пишет в правильный task_id (перессия критична). Нет auto-persist до Accept (AC101). Нет константы 8 в happy-path при наличии аналогов. Insufficient data row при пустой истории. Leave-one-project-out в analog retrieval. ВАС не влияет на similarity в initial estimate. Schema validation LLM features; malformed → fallback hierarchy. Sum(P80 tasks) ≠ project P80 (явный контракт агрегации).

19. Вне скоупа AI №1

Табель, anomalies, ETC, allocation людей, what-if, NL — отдельные контуры того же ProjectAIService; переиспользуют task/project embeddings и feedback store.