

# PET 3.0 - 智能项目估算平台

核心价值：让每一次项目估算都有据可循、有迹可查、降本增效

PET 团队 · Dong Wang

# 议程

## 业务价值 (2分钟)

1. 痛点与业务场景分析
2. 解决方案与落地路径

## 技术方案 (3分钟)

3. 技术栈 & 整体架构概览
4. AI Epic生成技术架构
5. Cost Breakdown功能技术原理

## 演示效果 (4分钟)


6. Live Demo
7. 功能完成度展示


## 团队与Q&A (1分钟)

8. 团队分工
9. Q&A

# 为什么需要 PET 3.0?

维度

 传统痛点

 PET 3.0 方案

数据管理

Excel/邮件/Smartsheet 分散, 历史难追溯

统一平台, 项目全生命周期可追踪

估算一致性

拍脑袋估算, 估算偏差大

标准化模板 + 统一费率, 口径一致

解释成本

"50人天怎么来的?" — 没人说得清

公式引擎逐行计算, **每一步可解释**

效率

Epic 拆分 xx 天, 工时计算靠手算

AI 生成 Epic **分钟级**, 公式引擎**秒级**计算

**Before**

黑盒 · 低效 · 不可控



**After**

透明 · 高效 · 可解释



# 业务场景：项目成本估算全链路



AI 加持



规则引擎

← 核心创新点

环节	当前方式	PET 3.0	核心提升
元数据	Excel 分散维护，无权限	管理员统一配置，全局生效	集中管控
项目管理	SmartSheet 多平台	统一平台，一站式管理	平台统一
BRD 录入	Excel 手动维护	系统结构化录入	数据结构化
Epic 录入	人工逐条录入	AI 智能生成，PM 仅需 Review	AI 自动化
工时估算	拍脑袋，Excel 手算	公式引擎自动计算	规则引擎
OGM 估算	Excel 手工填写	表单化填写，体验友好	体验提升
报表生成	透视表，样式简陋	实时生成，可视化报表	实时可视化



# 技术栈 & 整体架构概览

## 🔧 技术栈

- 前端: React + TypeScript + Tailwind
- 后端: Node.js (Next.js API Routes) + PostgreSQL
- 运维: Github + Vercel 自动部署

## 🤖 AI 能力

- DeepSeek API (JSON Mode)
- AI Coding 助手: **Cursor**、**Codex** 提升研发效率

## 整体架构



### 客户端

表单渲染 · 结果展示 · 可视化报表



### 服务端

Server API · 元数据配置 · 规则引擎等



### AI 层

DeepSeek API(Epic 生成)



### 存储层

PostgreSQL (角色/模板/公式/常量/估算记录等)



# AI Epic 生成技术架构

## 系统架构

### 用户输入需求描述

选择 BRD · 输入功能需求 · 快速模板



### Prompt Engineering

BRD 上下文注入 · 结构化输出约束 · JSON Schema



### DeepSeek 大模型

硅基流动 API · JSON Mode



### 结构化 Epic 输出

5-8 个 Epic · 优先级/功能点 · 批量入库

## 技术亮点

### 结构化 Prompt 设计

- 注入 BRD 标题、层级、现有 Epic 作为上下文
- 严格 JSON Schema 约束输出格式
- 预设生成要求：标题 10-30 字、描述 50-150 字

### 输出质量保障

- `response_format: json_object` 强制 JSON 输出
- 服务端二次校验 + 类型断言

 **30秒生成 5-8 个专业 Epic，效率提升 10x**



# Cost Breakdown：数据模型设计

## 多层配置体系

```
Team (团队)
├── Role (角色)
│   └── TaskTemplate (任务模板)
│       ├── Parameters[] (参数定义)
│       ├── Formula (公式)
│       │   └── Constants[] (常量)
│       └── category: shared | epic
```

## 核心表结构

表名	作用
task_templates	角色任务模板，关联公式
parameters	参数定义（类型/约束/默认值）
formulas	公式表达式 + 计算引擎
formula_constants	可配置常量（可编辑标记）
epic_estimations	估算结果 + breakdown

## 设计亮点

### ⚙️ 配置与代码分离

- 新增任务类型只需配置数据库，无需发版
- 公式、常量、参数约束均可热更新

### 📝 完整估时链路

```
epic_estimations (
  inputs JSONB,      -- 用户原始输入
  breakdown JSONB,   -- 计算过程快照
  computed_hours,    -- 最终结果
  created_by,        -- 操作人
  created_at         -- 时间戳
)
```

### 💡 任意历史估算都可完整复现计算过程



# Cost Breakdown: 动态公式引擎

## 核心架构

### 用户输入参数表单

选择Epic、Role、taskTemplate · 输入参数



### InputValidator 参数校验层

类型校验 · 范围约束 · 精度控制



### FormulaEngine 公式解析执行层

expr-eval 解析器 · 变量注入 · 数学运算



### Breakdown 计算过程保存

原始公式 · 解析表达式 · 结果快照

## 技术亮点

### 可配置公式引擎

- 公式存储在数据库，无需改代码即可调整计算逻辑
- 支持用户输入、常量两种类型参数
- 支持复杂表达式

### 多层参数校验

- 类型转换 + 范围约束 + 步长验证 + 枚举匹配

### 计算过程快照示例

```
{
  "expression": "page_count * base_hours",
  "resolved": "5 * 16",
  "result": 80,
  "unit": "hour"
}
```

- ✅ 解释时可精确还原每一步计算





# 功能完成度：已实现模块一览



## 用户管理

- 登录认证
- 用户 CRUD
- 用户统计



## 元数据管理

- Foundation Epic CRUD
- 核心参数配置



## 项目管理

- 项目信息 CRUD
- 搜索/筛选/分页
- 批量状态更新



## BRD & Epic

- BRD&Epic CRUD
- AI 生成 Epic
- dashboard



## Cost Breakdown

- 动态参数表单
- 公式引擎解析
- CRUD



## OGM 估算

- 工时自动汇总
- 多维度统计
- CRUD



## 预算报表

- 可视化图表
- BRD&EPIC Summary
- Budget Summary



## 技术基座

- CI/CD 自动化
- DataBase 集成
- 云服务器部署集成

7 大业务模块 + 完整技术栈 · 100% 可运行 · 演示环境已部署



# Live Demo



## PET 3.0 在线演示

<https://pet30.vercel.app/>



演示环境已部署，欢迎体验完整功能流程



# 团队分工



**Jeff Bu**

Delivery Manager

跨部门沟通/需求管理/整体进度  
把控



**Dong Wang**

Tech Lead

架构设计 / Cost Breakdown模  
块开发



**Maggie Zhu**

Full Stack

项目管理、Opex、报表等模块  
开发



**Summer Zhang**

Scrum Master

组织会议、跨部门沟通



**Groot Zhang**

Technical support

提供专业建议

🙏 谢谢聆听



用结构化数据，让项目估算标准快捷

PET 3.0 - 智能项目估算平台

Q & A 环节