

统一进度反馈组件

项目前分析报告

产品体验优化 · 任务状态反馈 · 立项决策 · MVP 规划 · 风险评估

产品类型: 统一进度反馈组件 / 耗时任务体验优化 / 任务状态反馈系统

文档用途: 项目前分析报告 / 立项评审材料 / 产品决策前置分析

版本日期: 2026 年 6 月

输出用途: 个人作品集项目资料 / 产品文档能力展示 / Codex 后续页面生成素材

文档信息

文档名称	统一进度反馈组件 项目前分析报告
项目名称	统一进度反馈组件
文档类型	项目前分析报告 / 立项评审材料 / 产品决策前置分析
版本	v1.0 统一排版版
日期	2026 年 6 月
产品类型	体验优化组件 / 基础产品能力 / 任务状态反馈系统
适用阶段	立项评审、产品决策、PRD 前置分析、作品集展示

重要说明

本文件将“进度条 UI”上升为“统一任务状态反馈能力”，重点分析等待场景中的用户不确定性、业务影响、组件化价值、MVP 范围、风险控制与立项建议。

本文件已按照《AI 体检报告解读与健康行动助手 AI 工作流与 Prompt 设计文档》的商务文档结构进行统一排版：封面、文档信息页、目录、分级标题、表格样式、正文行距、页脚与整体视觉层级保持一致。

目录

1. 报告摘要
2. 项目背景
3. 问题定义
4. 用户与场景分析
5. 用户痛点与需求分层
6. 数据现状与问题证据
7. 竞品与行业案例分析
8. 项目目标
9. 解决方案方向
10. 可行性分析
11. 收益评估
12. 成本评估
13. 风险分析与应对策略
14. MVP 范围建议
15. 阶段规划
16. 排期建议
17. 成功指标
18. 立项判断
19. 后续工作建议
20. 最终结论

1. 报告摘要

1.1 项目一句话定义

统一进度反馈组件，是一套面向产品内上传、下载、提交、导出、AI生成、视频转码、审核等待等耗时任务场景的通用状态反馈能力，用于让用户在等待过程中清楚知道系统是否响应、任务是否进行、当前处于哪个阶段、是否成功、失败后如何处理。

1.2 核心问题

当前产品中存在多个耗时任务场景，但用户在等待过程中缺少清晰、统一、可信的进度反馈。用户点击操作后，常常无法判断系统是否已响应、任务是否卡住、还需要等待多久、是否可以关闭页面、失败后应该如何处理。

- 用户等待焦虑增加。
- 用户重复点击、重复提交、刷新页面或强制退出。
- 上传、生成、导出等任务完成率下降。
- 客服中关于“卡住”“没反应”“是否失败”的咨询增加。
- 产品整体显得不稳定、不专业，影响用户信任。
- 不同业务模块加载样式不统一，造成体验割裂。

1.3 项目建议

建议立项，优先级建议为 P1。本项目表面是 UI 体验优化，实质是对产品中“等待场景”和“后台任务状态”的系统性治理。它不只是增加一条进度条，而是建立一套可复用的任务状态反馈机制。

- 文件上传。
- 数据导出。
- AI 内容生成。
- 视频转码 / 内容处理。
- 表单或订单提交。

1.4 预期收益

- 降低任务中途退出率。
- 降低重复点击和重复提交行为。
- 提高上传、生成、导出等任务完成率。
- 减少用户因等待不确定产生的负面反馈。
- 降低客服和运营解释成本。
- 提升产品可靠感、专业感和一致性。
- 形成可复用基础组件，降低后续业务重复开发成本。

1.5 结论

建议以 MVP 方式启动该项目。第一阶段不追求覆盖所有复杂任务，而是先完成基础进度反馈、按钮 loading、失败重试、阶段型反馈、基础埋点能力。验证核心指标后，再扩展后台任务、任务中心、通知提醒、历史任务记录等高级能力。

2. 项目背景

2.1 产品背景

在现代数字产品中，用户越来越频繁地面对需要等待系统处理的任务。尤其在文件上传、文件下载、视频上传与转码、数据导出、报告生成、AI 文本/图片/视频生成、表单提交、订单提交、支付确认、内容审核、云端同步和批量处理等场景中，等待不可避免。

这些任务的共同特点是：用户完成前台操作后，系统需要在后台继续处理。此时，如果界面没有及时、明确、持续的反馈，用户就会进入“不确定等待”状态。用户不一定不能接受慢，但很难接受不知道系统是否在正常工作。

2.2 用户体验背景

用户在等待中通常会产生以下心理问题：

- 我刚才点成功了吗？
- 系统是不是卡住了？
- 为什么没有任何变化？
- 还要等多久？
- 我能不能关闭页面？
- 我是不是应该再点一次？
- 失败了怎么办？
- 我的文件、订单、内容会不会丢？

这些问题如果没有被产品界面回答，用户就会用自己的猜测来解释系统状态。通常情况下，用户不会默认相信系统仍在正常工作，而是倾向于认为产品卡住、出错或不稳定。

2.3 业务背景

从业务角度看，等待反馈不足会影响关键流程完成率。上传流程中用户可能因看不到进度而退出页面；AI 生成场景中用户可能因长时间等待而取消任务；数据导出场景中用户可能重复点击导出按钮，造成重复任务；支付或订单提交场景中用户可能重复提交，造成订单异常。

所以该项目并不只是视觉优化，而是会影响任务完成率、用户信任、运营成本和系统稳定性。

3. 问题定义

3.1 核心问题

当前产品在耗时任务场景中缺少统一、明确、可信的任务进度反馈机制，导致用户无法理解系统状态、无法管理等待预期、无法在异常发生时获得有效处理路径。

3.2 问题表达

在上传、生成、导出、提交、审核等耗时任务场景中，用户完成操作后，因为系统缺少清晰的进度反馈和状态说明，无法判断任务是否正常进行，进而产生等待焦虑、误操作、中途退出和信任下降。

3.3 问题拆解

3.3.1 操作反馈不足

用户点击按钮后，如果按钮没有变化、页面没有提示，用户会怀疑操作没有生效。典型表现包括重复点击按钮、重复提交表单、重复上传文件、反复刷新页面、误以为系统无响应。

3.3.2 进度不可见

对于较长任务，如果用户只能看到“加载中”，而看不到任务阶段或进度，会难以判断等待是否值得继续。用户不知道任务完成多少、不知道是否卡住、不知道还要等多久，最终可能中途退出。

3.3.3 状态文案不清楚

很多产品只显示“处理中”“加载中”，但用户并不知道具体在处理什么。更好的方式应该告诉用户正在上传文件、正在校验文件、正在生成内容、正在导出报告、正在保存结果或正在完成最后处理。

3.3.4 异常处理不足

任务失败、超时、断网、权限不足、文件格式错误等情况如果没有明确提示，会让用户无法继续操作。典型问题包括失败原因不明确、没有重试入口、没有取消入口、没有恢复路径，用户只能退出或联系客服。

3.3.5 体验不统一

不同模块各自实现 loading、弹窗、进度条和错误提示，会造成体验割裂。表现为有的页面有进度条、有的没有；有的页面可以取消、有的不能；有的失败提示详细、有的只显示“失败”；不同页面视觉样式不统一；后续维护成本高。

4. 用户与场景分析

4.1 目标用户

用户类型	典型行为	主要痛点
------	------	------

普通用户	提交表单、上传资料、下载文件	不知道是否提交成功，不知道是否还在处理
内容创作者	上传视频、发布内容、等待审核	担心作品上传失败、转码失败、内容丢失
AI 功能用户	提交生成任务、等待结果	不知道 AI 是否还在生成，不知道失败后怎么办
企业后台用户	导出数据、批量处理任务	等待时间长，缺少后台任务状态
运营人员	审核内容、批量发布、处理任务	多任务状态不清楚，异常处理成本高
新用户	初次使用产品功能	对系统缺乏信任，更容易因无反馈流失

4.2 高频场景

场景一：文件上传

用户上传文件后，需要等待上传完成。如果没有上传进度，用户会担心文件是否正在上传、网络是否中断、页面是否卡住。用户需要看到上传百分比、文件大小与上传速度、上传是否成功，并在失败后可以重试，过程中可以取消。

场景二：AI 生成

用户提交 AI 生成任务后，系统可能需要较长时间生成结果。由于 AI 生成过程不一定能准确百分比化，所以用户更需要阶段型反馈，例如已接收任务、正在理解、正在生成、正在优化或正在保存。

场景三：数据导出

用户点击导出后，系统需要生成文件。如果导出过程没有反馈，用户可能重复点击导出按钮，造成多个重复任务。用户需要知道导出任务已创建、文件正在生成、生成完成后可下载、失败后知道原因，长任务可以后台运行。

场景四：订单或表单提交

用户提交表单、订单或支付相关操作时，系统需要等待服务端确认。如果没有反馈，用户可能重复提交，造成数据异常。点击后按钮应立即变为提交中，提交中禁止重复点击，成功后展示明确成功状态，失败后展示可理解的错误原因。

场景五：视频转码 / 内容处理

视频上传完成后，系统仍然需要转码、生成封面、审核或保存。此时如果进度条直接显示 100%，但用户还不能使用视频，会造成误解。应区分“上传完成”和“处理完成”，明确当前处于转码、生成封面还是审核阶段。

5. 用户痛点与需求分层

5.1 用户痛点

痛点	用户表现	产品影响
不知道操作是否成功	重复点击、重复提交	系统压力增加，数据异常
不知道系统是否卡住	刷新页面、退出页面	任务失败率上升
不知道还要等多久	焦虑、放弃等待	完成率下降

不知道失败原因	无法自助解决	客服咨询增加
不知道能否离开页面	被迫等待	用户体验下降
不知道下一步做什么	流程中断	转化率下降

5.2 需求分层

层级	需求内容
基础需求	操作后立即有反馈；系统处理中有 loading；任务成功后有成功提示；任务失败后有失败提示。
进阶需求	可计算任务展示百分比；长任务展示当前阶段；失败后支持重试；用户可以取消任务；长任务可以后台运行。
高级需求	多任务统一管理；任务完成后通知；任务历史记录；失败任务恢复；跨页面状态同步；多端任务状态同步。

本项目第一阶段重点满足基础需求和部分进阶需求，不直接进入复杂任务中心建设。

6. 数据现状与问题证据

6.1 需要收集的数据

- 上传任务开始次数、完成次数、失败次数。
- 上传过程中页面退出次数。
- 上传按钮重复点击次数。
- 导出按钮重复点击次数。
- AI 生成任务取消次数和失败次数。
- 等待页面平均停留时长。
- 用户刷新页面次数。
- 客服反馈中相关关键词数量。
- 用户反馈中“卡住”“没反应”“慢”“失败”等关键词数量。

6.2 关键判断指标

指标	说明
任务完成率	衡量用户是否成功完成任务
中途退出率	衡量等待过程中用户是否放弃
重复点击率	衡量用户是否认为操作未生效
失败率	衡量任务技术或流程稳定性
重试成功率	衡量失败后的恢复能力
客服咨询量	衡量用户是否能自助理解状态
等待页停留时长	衡量用户等待成本
取消率	衡量任务是否让用户失去耐心

6.3 数据分析假设

- 等待时间越长，中途退出率越高。
- 无按钮 loading 的表单，重复点击率更高。
- 没有失败原因的任务，客服咨询更多。
- AI 生成类任务如果只有“生成中”，用户取消率更高。
- 上传完成和处理完成没有区分时，用户更容易误以为系统异常。
- 进度条长时间静止会比没有进度条更容易破坏信任。

6.4 数据结论

即使当前没有完整量化数据，仅从用户体验和任务流程角度看，该项目仍然具备较高建设价值。因为所有需要等待的关键流程，都天然需要系统状态反馈。但为了验证项目效果，上线前必须补充基础埋点，上线后通过数据对比判断项目价值。

7. 竞品与行业案例分析

7.1 分析对象

- 网盘产品：百度网盘、Google Drive、Dropbox、OneDrive。
- 视频平台：YouTube、Bilibili、抖音创作者后台。
- AI 产品：ChatGPT、Midjourney、Runway、可灵、即梦。
- 电商产品：淘宝、京东、Amazon。
- 外卖物流：美团、饿了么、Uber Eats、DoorDash。
- 办公软件：Notion、Figma、Canva、飞书、钉钉。
- 系统软件：Windows、macOS、iOS、Android。

7.2 行业通用做法

类别	常见做法	可借鉴点
上传下载类	百分比、已完成大小/总大小、速度、剩余时间、暂停/取消/重试、后台上传	任务可计算时百分比最清晰；大文件任务需要取消和重试能力；上传完成后仍需处理时，要明确区分处理阶段。
AI 生成类	动态生成中状态、阶段型文案、预计等待提示、失败重试、队列位置	AI 任务不适合强行展示虚假百分比；阶段型反馈比单一 loading 更可信。
物流配送类	阶段节点、当前位置、预计送达时间、异常提示	对不可连续量化的流程，可以用阶段节点表达进度。
游戏成长类	经验条、任务完成度、距离下一奖励、成就进度	进度不只是反馈，也可以成为激励机制。

7.3 竞品分析结论

成熟产品普遍不会让用户在长任务中只看到空白页面或单一“加载中”。它们会通过百分比、阶段节点、状态文案、预计时间、取消重试、后台运行等方式，让用户理解任务状态。因此，本项目不应被理解为单个 UI 组件，而应该被定位为“任务进度反馈系统”的基础建设。

8. 项目目标

8.1 用户目标

- 用户操作后能够立即获得反馈。
- 用户等待过程中能够知道任务正在进行。
- 用户能够理解当前任务处于哪个阶段。
- 用户能够判断是否继续等待。
- 用户在失败时能够知道原因和下一步操作。
- 用户对产品稳定性和可靠性的感知提升。

8.2 业务目标

- 提高上传、生成、导出等任务完成率。
- 降低等待页面中途退出率。
- 降低重复点击率和重复提交率。
- 降低任务失败后的用户流失。
- 减少相关客服咨询。
- 提升核心流程转化率。
- 建立可复用组件能力。

8.3 产品目标

- 统一全产品等待反馈样式。
- 规范不同任务类型的反馈方式。
- 建立标准任务状态体系。
- 建立异常状态处理规范。
- 建立进度反馈埋点体系。
- 为后续任务中心、通知中心、后台任务能力打基础。

8.4 数据目标

- 任务完成率提升。
- 中途退出率下降。
- 重复点击率下降。

- 失败后重试成功率提升。
- 相关客服咨询下降。
- 用户负面反馈下降。
- 等待页停留行为更稳定。

9. 解决方案方向

9.1 总体方案

建设一套统一进度反馈组件，根据任务类型和耗时长短，采用不同反馈方式。核心不是所有场景都用同一个进度条，而是根据任务特征选择合适反馈。

- 短任务：按钮 loading。
- 中等任务：loading + 状态文案。
- 可计算长任务：百分比进度条。
- 不可计算长任务：阶段型进度。
- 超长任务：后台运行 + 完成通知。
- 异常任务：错误原因 + 重试 / 取消 / 查看详情。

9.2 任务类型分层

任务类型	等待时间	反馈方式
即时任务	1 秒以内	按钮状态变化
短等待任务	1-3 秒	按钮 loading / 轻量 loading
中等待任务	3-10 秒	loading + 状态文案
可计算长任务	10 秒以上	百分比进度条
不可计算长任务	10 秒以上	阶段型进度
超长任务	30 秒以上	后台运行 / 通知 / 任务状态

9.3 功能模块

模块	适用场景	核心能力
按钮反馈	短任务，如提交、保存、确认	点击后按钮变为 loading；按钮禁用；成功进入下一步；失败恢复按钮并提示错误。
普通 loading	中等任务，如页面加载、短请求处理	展示加载动画；展示当前状态文案；超时后进入异常状态。
百分比进度条	上传、下载、文件处理等可计算任务	展示百分比、任务名称、当前状态、剩余时间；支持取消、重试。
阶段型进度	AI 生成、视频转码、审核等不可精确计算任务	展示当前阶段和阶段说明；长时间停留时展示补充说明；成功或失败后展示结果。
异常处理	失败、超时、断网、权限不足、文件过大、格式不支持	展示错误原因、解决建议、重试、取消、返回、联系客服等入口。
数据埋点	所有接入场景	记录任务开始、进度展示、进度更新、成功、失败、取消、退出、重试等事件。

10. 可行性分析

10.1 产品可行性

本项目具备较高产品可行性。原因包括问题场景明确、用户痛点普遍存在、组件可复用价值高、可以分阶段上线、可以通过数据验证效果，并且不依赖一次性完成所有复杂功能。

10.2 技术可行性

可计算任务如上传、下载，通常可以通过已完成字节数 / 总字节数计算进度，技术可行性较高。不可计算任务如 AI 生成、视频转码、审核，难以准确计算百分比，建议通过后端返回任务阶段，前端展示阶段型反馈。后台任务如果支持后台运行、任务通知、任务历史，则需要后端任务队列、任务状态查询、通知系统等支持，建议放在后续阶段。

10.3 资源可行性

第一阶段资源投入可控。建议投入产品 1 人、UI/UX 设计 1 人、前端 1-2 人、后端 1 人、测试 1 人、数据分析 0.5 人。如果只做基础组件和 2-3 个核心场景接入，项目周期预计 4-6 周。

10.4 运营可行性

该项目对运营侧有正向价值。它可以减少用户咨询、降低重复解释成本、帮助运营判断任务失败原因、改善内容发布/审核/导出等后台流程，并提高创作者和企业用户体验。

10.5 成本可行性

项目成本主要来自组件设计、前端组件开发、后端任务状态接口、不同业务场景接入、异常状态测试和数据埋点。但由于该组件具有高复用价值，长期看投入产出比高。

11. 收益评估

11.1 用户收益

- 等待过程更清楚。
- 操作反馈更及时。
- 不再因为界面无变化而焦虑。
- 失败后知道如何处理。
- 长任务可选择后台运行。
- 对产品稳定性更信任。

11.2 业务收益

- 提升任务完成率。
- 降低中途退出率。
- 降低重复提交风险。
- 降低客服咨询。
- 提升关键流程转化。
- 提升用户对产品专业度的评价。

11.3 研发收益

- 减少各业务重复开发进度反馈。
- 统一组件规范。
- 降低后续维护成本。
- 统一异常状态处理。
- 建立标准任务状态协议。

11.4 组织收益

- 形成基础体验能力。
- 提升跨模块体验一致性。
- 为后续任务中心、通知中心打基础。
- 增强产品体验体系化建设能力。

12. 成本评估

12.1 一次性建设成本

成本类型	内容
产品成本	需求分析、状态设计、规则定义、验收标准
设计成本	组件样式、交互状态、动效、适配
前端成本	组件开发、页面接入、状态控制
后端成本	任务状态接口、失败原因返回、取消 / 重试支持
测试成本	正常流程、异常流程、兼容性、性能测试
数据成本	埋点设计、上报、看板分析

12.2 长期维护成本

- 组件版本维护。
- 新业务场景接入。
- 异常状态补充。
- 埋点字段维护。
- 任务状态协议升级。

12.3 机会成本

如果团队当前有更高优先级的商业化、增长或核心功能项目，需要评估该项目是否抢占资源。但考虑到本项目可作为多个业务模块的基础能力，建议将其视为基础体验建设，而不是单点优化。

13. 风险分析与应对策略

风险	风险说明	影响	应对策略
进度不准确	部分任务无法准确计算百分比，强行展示会造成误解	用户认为系统欺骗自己，降低信任	可计算任务才展示百分比；不可计算任务使用阶段型进度；上传完成但处理未完成时切换为“正在处理文件”。
组件接入范围过大	如果一开始覆盖所有业务模块，项目容易延期	需求膨胀，研发压力变大	第一阶段只覆盖 2-3 个高频场景；明确 MVP 范围；非核心能力放到后续版本。
后端状态接口不完善	前端依赖后端返回任务状态，如果状态不标准，只能做模糊展示	进度反馈不可信	建立统一任务状态枚举；先支持基础状态：处理中、成功、失败、超时、取消。
异常状态覆盖不足	如果只考虑成功流程，失败时体验仍然差	用户失败后无路可走，客服压力不降反升	PRD 中列完整异常清单；测试覆盖异常场景；失败提示必须包含原因和下一步操作。
上线后指标不明显	原始数据不完整或试点场景不典型，难以证明价值	项目价值难以评估	上线前补齐埋点；选择高频场景试点；做上线前后数据对比，必要时 A/B 测试。

14. MVP 范围建议

14.1 MVP 目标

第一阶段目标不是做完整任务中心，而是解决最核心的问题：用户点击后有反馈；用户等待时知道系统正在处理；可计算任务有进度；不可计算任务有阶段说明；失败后可以重试；基础数据可以被记录。

14.2 MVP 包含内容

- 按钮 loading。
- 普通 loading。
- 百分比进度条。
- 阶段型进度展示。
- 成功状态。
- 失败状态。
- 重试入口。
- 取消入口。

- 基础埋点。
- 2-3 个核心业务场景接入。

14.3 MVP 不包含内容

- 完整任务中心。
- 跨设备任务同步。
- 系统级通知。
- 任务历史列表。
- 多任务并行管理。
- 高级断点续传。
- 完整后台任务队列管理。

15. 阶段规划

阶段	目标	内容	适用场景
第一阶段：基础进度反馈	解决用户“不知道系统是否响应”的问题	按钮 loading、loading 状态文案、基础进度条、成功/失败状态、重试能力、基础埋点	表单提交、文件上传、数据导出
第二阶段：长任务体验优化	解决用户“不知道还要等多久、能不能离开”的问题	阶段型进度、预计剩余时间、取消任务、后台运行、长时间等待提示、异常状态细分	AI 生成、视频转码、大文件处理、批量任务
第三阶段：任务中心能力	解决多任务、后台任务、历史任务管理问题	任务列表、状态查询、完成通知、历史记录、失败恢复、多任务并行展示	企业后台、创作者中心、批量任务平台、高级 AI Agent 任务

16. 排期建议

阶段	时间	主要工作
需求分析	第 1 周	明确场景、状态、指标、MVP 范围
产品设计	第 1-2 周	输出 PRD、状态流程、埋点方案
交互设计	第 2 周	组件交互、异常状态、页面流程
UI 设计	第 2-3 周	组件视觉、动效、适配规范
前端开发	第 3-4 周	组件开发、页面接入
后端开发	第 3-4 周	状态接口、任务接口、错误码
测试联调	第 5 周	正常流程、异常流程、兼容性测试
灰度上线	第 6 周	小流量上线、数据观察
全量上线	第 7 周	全量发布、监控和复盘

17. 成功指标

17.1 北极星指标

耗时任务完成率。该指标最能体现项目价值。进度反馈最终不是为了让界面更好看，而是让用户更顺利地完成任务。

17.2 核心指标

- 任务完成率。
- 等待页面中途退出率。
- 重复点击率。
- 任务失败率。
- 失败后重试成功率。
- 用户取消率。
- 相关客服咨询量。

17.3 过程指标

- 进度组件曝光次数。
- 进度更新次数。
- 平均任务耗时。
- 平均等待页面停留时长。
- 取消按钮点击率。
- 重试按钮点击率。
- 后台运行点击率。

17.4 防御指标

- 页面性能不能明显下降。
- 错误率不能上升。
- 接口请求量不能异常增加。
- 前端渲染不能造成卡顿。
- 用户投诉不能增加。

17.5 参考目标

指标	预期方向
任务完成率	提升 5%-10%
等待页退出率	下降 10%-20%
重复点击率	下降 20%-30%
失败后重试成功率	提升 10%-15%
相关客服咨询	下降 10%-15%
用户负面反馈	下降

18. 立项判断

18.1 是否值得做

值得做。理由是：问题真实存在、用户痛点明确、适用场景广泛、业务收益可验证、组件复用价值高、技术风险可控、可分阶段推进，并且对产品可靠感有明显提升。

18.2 是否现在做

建议现在做。AI 生成、上传、导出等长任务场景越来越多，用户对产品反馈速度和透明度要求越来越高。如果不做统一组件，各业务会继续重复开发。越早建立统一标准，后续维护成本越低。

18.3 优先级建议

建议优先级：P1。如果公司当前资源紧张，也至少应作为 P2 高优先级体验基础建设推进。

18.4 推荐立项方式

建议采用“小范围 MVP + 数据验证 + 分阶段扩展”的方式立项。第一阶段不追求大而全，避免变成复杂任务平台。先做高频等待场景，验证价值后再逐步扩展。

19. 后续工作建议

明确第一阶段接入场景。

补充现有数据埋点。

输出详细 PRD。

与研发确认任务状态协议。

与设计确认组件视觉规范。

与测试确认异常状态清单。

灰度上线核心场景。

对比上线前后数据。

输出项目复盘报告。

决定是否进入第二阶段。

20. 最终结论

统一进度反馈组件项目，本质上不是一个单纯的进度条 UI 项目，而是一次面向耗时任务场景的产品体验基础建设。

它解决的核心问题是：用户在等待系统处理任务时，无法理解系统状态、无法判断等待预期、无法获得异常处理路径。

从用户角度看，该项目可以降低等待焦虑，增强控制感和信任感。从业务角度看，该项目可以提升任务完成率，降低中途退出和重复操作。从研发角度看，该项目可以形成统一组件，减少重复开发和维护成本。从长期产品能力看，该项目可以为任务中心、后台任务、通知系统、AI Agent 执行流等能力打基础。

因此，建议该项目进入立项流程，优先以 MVP 方式启动，第一阶段聚焦上传、导出、AI 生成等高频长耗时场景，验证核心指标后再逐步扩展至更多业务模块。

最终立项建议	内容
项目名称	统一进度反馈组件
项目类型	体验优化 / 基础组件建设
建议优先级	P1
推荐方式	MVP 先行，灰度验证，分阶段扩展
核心目标	提升耗时任务完成率，降低等待不确定性，建立统一任务状态反馈能力