PRISM

プロジェクト可視化・自動化プラットフォーム / 企画案 1枚サマリー

株式会社リベラウェア 和田様 ご提案
2026-07-30 作成 / TEKION Group
詳細資料：企画ドキュメント 7本・提案デッキ 12枚

動いているプロジェクトの「いま」が、ひとりでに見える。

1 なぜ既存ツールでは駄目なのか

導入直後	1ヶ月後	3ヶ月後	6ヶ月後
全員が律儀に入力	**忙しい人から遅れる**	**更新が週1になる**	**誰も見ない墓場**
運用ルールが守られている	最も情報を持つ人が最初に落ちる	見ても実態と合っていない	そしてツールを乗り換える

Jira・Backlog・Notion・Asana・Redmine ― すべて優れたツール。共通するのは **「人間が正しくステータスを更新し続ける」** という前提。 少人数が多数のプロジェクトを並行して回す環境では、この前提が真っ先に崩れる。 **ツールが悪いのではなく、アーキテクチャが環境と構造的に相性が悪い。**

2 PRISM の3本柱

柱 01
Zero-Input 進捗管理
GitHub・Slack・カレンダー・Drive・見積請求・現場報告。すでに存在する6系統のログをAIが解釈して進捗を推定する。 人間の仕事は「入力」ではなく **「推定が違っていたときの訂正」** だけ。訂正はそのまま学習データになる。

柱 02
3階層への自動翻訳
同じ実態を、見る人の役割に応じて自動で言い換える。 **下の階層に「上への報告のための作業」を一切発生させない。** 経営層のダッシュボードは、開発者が何もしなくても埋まる。

柱 03
プロセス分解と自動化提案
蓄積された実作業ログから、**誰も書いていないが実在するプロセス**を再構成。 滞留・手戻り・属人化を自動検出し、「月8回の定型報告書を自動生成しますか？」まで提案する。

3 同じ実態を、3つの粒度で

経営層
全 27 案件中、赤信号 3 件。うち2件は同じ人のリソース超過が原因。来月の売上見込みは確度 87 %― 意思決定に必要な粒度。数字と信号と選択肢。

PM
案件Aは設計フェーズで 3.2 日遅延。原因は仕様確定待ち。田中さんが今週3案件を掛け持ち、稼働 132 %― 段取りに必要な粒度。ボトルネックと打ち手。

開発者
今日やるべきは 3 つ。うち1つは他の人の待ちを生んでいるので優先。報告は不要、勝手に上がる ― 手を動かすのに必要な粒度。今日の一手だけ。

4 競合との違い

	既存のPMツール	BIツール	PRISM
進捗データの入手	人間が入力	既存DBから取得	**AIが自動推定**
3階層への翻訳	手作業で集計	ダッシュボード自作	**自動**
属人化の可視化	なし	なし	**バス係数を算出**
自動化への接続	外部連携のみ	なし	**提案 → 実装まで伴走**
導入負荷	高い（設計が必要）	高い	**低い（繋ぐだけ）**

PRISM は既存ツールの置き換えを求めない。 Jira を使い続けたまま、その上に乗るレイヤーとして読む。導入判断のハードルが劇的に下がる。

5 事業性【仮説】

1,500円/人
Standard 月額。Starter 800円/人、Enterprise は個別見積。自動化実行数は従量課金

90万円
想定 ARPA（50名 × 1,500円 × 12ヶ月）

5.6
LTV / CAC。SaaS健全基準 3.0 を満たす。CAC回収 10.7ヶ月

25万円/月
12名規模で現在支払っている報告コスト（置き換え原資）

6 最大のリスクと、その潰し方

- **R1 推定精度が実用に耐えない（致命的）** → Phase 0 で実データ検証を最優先。確信度を必ず付与し、閾値未満は「要確認」として人間へ回す
- **R2 連携できないデータがある（発生確度：高）** → 一言報告ボット＋Excel取込。「全部繋がらなくても、繋がった分だけ価値が出る」設計
- **R3 監視ツールだと受け取られる** → 個人の生産性指標を**意図的に実装しな**
- **R4 既存ツールとの共存** → 置き換えを求めない。Jira の上に乗るレイヤ