プロジェクトは「作る」活動ではない。

合意を設計する活動である。

Agreement-Driven Development
（ADD｜合意駆動開発）

AI時代の上流工程を再定義する、新しいプロジェクト成功の方程式

株式会社八重洲テックソリューションズ ｜ Decision OS

なぜプロジェクトは炎上するのか？

契約している。でも、合意していない。　—— これがデスマーチの正体です。

RFPが曖昧

解釈がバラバラで共通認識がない

完成イメージがない

画面・業務フローが共有されていない

解釈がバラバラ

同じ文書を見て違うものを想像している

見積が感覚依存

根拠が説明できず、炎上リスクを内包

合意が成立していない状態で契約 → 開発開始 → 「それ違う」 → 手戻り・炎上・追加コスト

ADDとは何か？　―― プロジェクト開始条件を定義する方法論

ADD（Agreement-Driven Development）= 合意を構造化し、可視化し、検証し、定量化してから契約・開発を開始する方法論

① **解釈排除**
No Ambiguity

誰が見ても
同じ意味になる

② **可視化**
Visualization

画面・動線・振る舞いが
共有されている

③ **検証**
Validation

ステークホルダー全員が
同じ理解である

④ **定量化**
Quantification

見積・工数・リスクが
数値で説明できる

この4つが満たされて初めて「合意が成立した」と定義し、契約・開発へ進む。

ADDライフサイクル（7つのフェーズ）

ADDとPoCの違い　―― 目的が違う。検証するものが違う。順番が違う。

PoC（Proof of Concept）

技術的に「できるか」を検証する

目的：技術的実現可能性の確認

対象：技術・ツール・アーキテクチャ

タイミング：開発前 or 開発中

成果：技術的可否

失敗の意味：技術選定ミス

PoC ＝「できるか」

ADDのプロトタイプ

「これで合意できるか」を検証する

目的：合意の成立確認

対象：画面・業務フロー・振る舞い・操作

タイミング：契約前（超上流）

成果：認識一致・合意成立

失敗の意味：契約ミス・炎上リスク

ADD ＝「これでいいか」

PoCは技術を保証する。　ADDはプロジェクトを守る。
両方必要だが、順番がすべて。「合意を先に。技術はその後。」

AIの本来の役割　―― 合意を成立させるための「補完エンジン」

✕ 順序を間違えると、AIはリスクになる

→ 曖昧な要求

→ AIが解釈・補完（方向がズレる）

→ 誤った仕様・コードが高速生成

- **手戻り・炎上加速**

AIは、前提が曖昧なままだと「間違いを加速する」

☑ 正しい順序：合意を構造化してからAIを活用

①合意設計（ADD）

②AIで補完・言語化（空白を埋める）

③合意検証（認識の一致）

④契約 → ⑤PoC → ⑥開発

この順番で初めて、AIは価値を最大化する

ADDが「構造」を作り、AIが「空白」を埋める。　AIは開発を変えるのではない。合意の精度を変える。

空白を補完する

不足している要求や考慮事項を提案する

理解を深める

曖昧な点を掘り下げ、前提や意図を明確にする

言語化する

朧げな要求を、構造化された言葉と仕様に変換する

創造AIが来ても、ADDは不要にならない　―― 勝負の場所が「生成」から「統制」へ

創造AI（生成レイヤー）

- ✓ 業務フローを理解し、高品質なプロトタイプを生成
- ✓ 画面・遷移・例外まで自動生成
- ✓ 複数案を高速で提示

ここは競争しない領域（コモディティ化）

価値：スピード・量・創造性

×

ADD（統制レイヤー）

- ✓ 採用基準（Acceptance Criteria）の固定
- ✓ 契約条項への拘束・トレーサビリティ
- ✓ 例外・境界の完全性保証
- ✓ 変更の影響分析→承認→一貫制御

価値：精度・再現性・責任

AIは仕様を「作る」　ADDは仕様を「効かせる」
プロジェクトの成功は「決めて、守り、回す」で決まる

段階導入モデル　―― 今すぐ価値を出し、ADDで最大化へ

フェーズ①

意思決定OS単体で導入

すぐに価値提供（ADDなしでもOK）

- EVM・KPI・可視化・予測・アラート
- プロジェクトの「今」を見える化
- ズレやリスクを早期に検知

★☆☆☆☆
価値：現状把握・問題発見

フェーズ②

一部のADDを導入

一部の構造化で精度が向上

- 画面構造・業務フロー・データ定義・ルール
- 一部のズレが解消され判断精度が向上
- 意思決定リスクをアラートが「当てる」ように

★★★☆☆
価値：実用レベルの判断が可能に

フェーズ③

ADDをフル適用

真の合意構造で最大価値を発揮

- 全体構造・例外定義・振る舞い・合意形成・契約
- 前提のズレがゼロに
- 意思決定OSが最大精度で機能

★★★★★
価値：ズレない判断・プロジェクト成功率が劇的向上

戦略：意思決定OSで入り、ADDで深く。アップセルの導線が自然に完成。

ADDがもたらす価値 —— 属人知を「検証された組織知」に変える

問題1：特定の人に依存する

その人がいないと回らない
判断が再現できない

再現性がなく、スケールしない

問題2：知識が蓄積・共有されない

聞いた内容が残らない
同じ質問が繰り返される

知識が資産にならない

問題3：正しいか検証されない

そのやり方は本当に正しいのか？
リスクや非効率が固定化する

リスクや非効率が固定化する

解決策：ADD（Agreement-Driven Development） 属人知を「検証された組織知」に変える仕組み

① 外に出す

知識を構造にする

② 検証する

ルール整合・効率性・リスク

③ 合意する

組織としてOKか？

④ 固定する

標準・ルールとして固定

⑤ 共有・改善

フィードバックで継続改善

最終的な姿：誰がやっても、同じ品質で、同じ判断ができる組織へ 「知識ではなく、構造で勝つ。」

まとめ —— ADDが解決する3つの問題

Q なぜ炎上するのか？

→ 合意が言葉でしかなく、構造化されていないから。ADDが合意を設計図に変える。

Q PoCだけで大丈夫か？

→ 技術はOKでも仕様がズレる。ADDで「これでいいか」を先に固める。

Q AIが来ても必要か？

→ AIは生成を加速するが、統制しなければ方向がズレる。ADDが構造を作り、AIが価値を最大化する。

見える化だけでは、変わらない。構造でしか、変えられない。

ADD × Decision OS = プロジェクト成功の方程式