



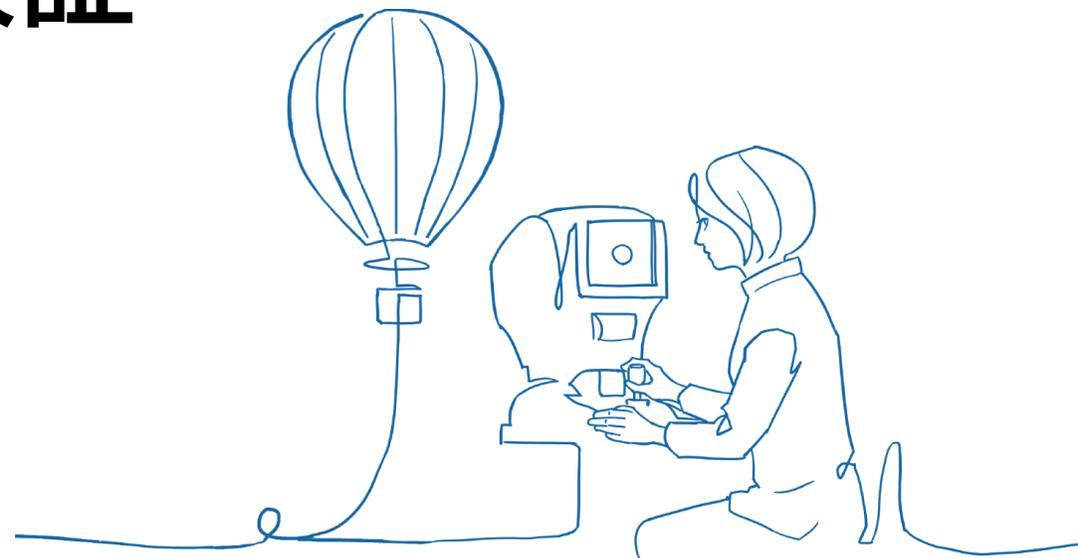
ソフトウェア生産性向上の可視化プロセス — AIエディターの効果検証 —

2025年10月22日

株式会社ニデック

開発本部 サージカル開発部 人工網膜開発課

山崎裕司



医療ソフトウェアの開発プロセス

医療機器ソフトウェアの安全性と有効性はプロセスにより品質を保証する。

医療機器の品質マネジメントシステム: ISO 13485
医療機器のリスクマネジメント: ISO 14971

医療機器の電気安全: IEC 60601-1
医療機器のユーザビリティエンジニアリング: IEC 62366-1

医療機器ソフトウェアのソフトウェアライフサイクルプロセス: IEC 62304



Class A	✓	✓			✓			✓	✓
Class B	✓	✓	✓		✓	✓	✓	✓	✓
Class C	✓	✓	✓	✓	✓	✓	✓	✓	✓

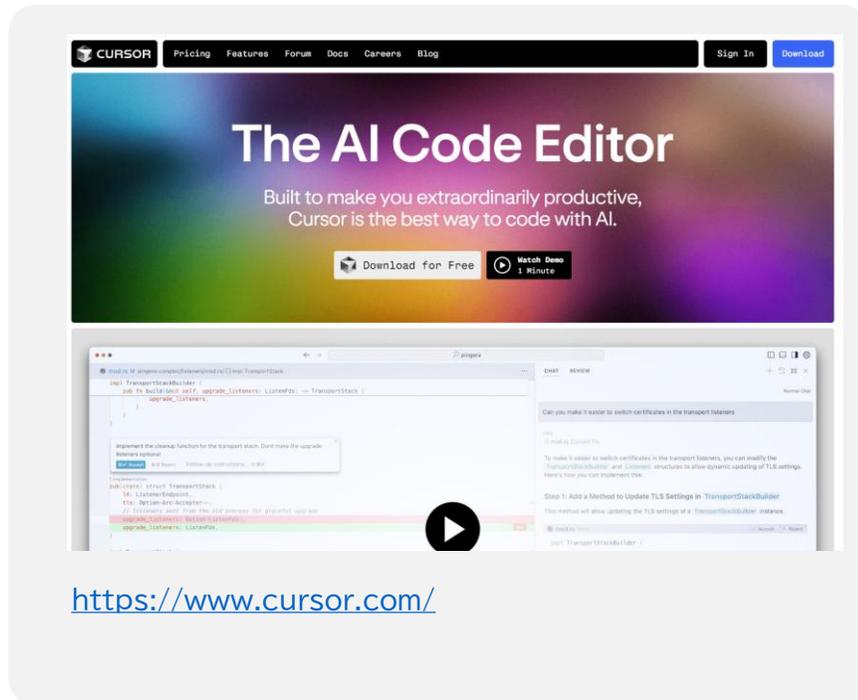
※手術機器のソフトウェアはクラスCに分類

ソフトウェア構成管理

ソフトウェア問題管理

2024年の取組み

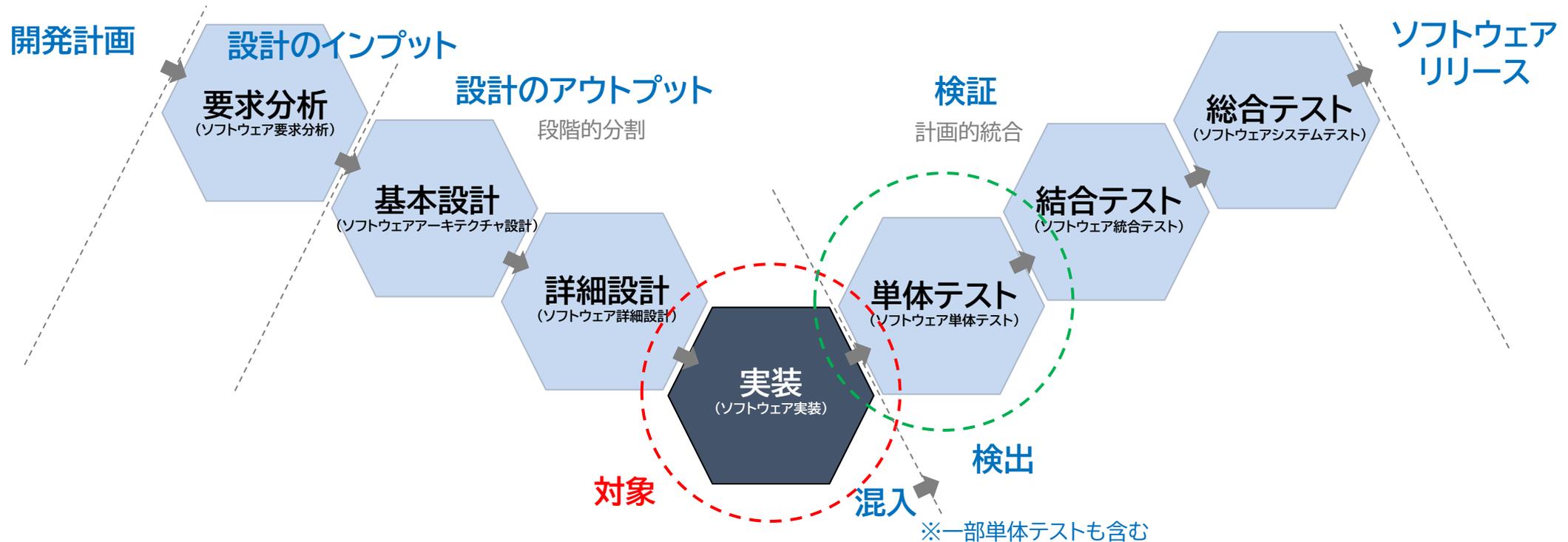
生成AI活用をコーディング工程に活用することで生産性向上と人材育成への期待を報告した。



トライアル設計

計画・準備:	2023年12月1日 ~	2023年12月27日
トライアル前半:	2024年1月10日 ~	2024年3月31日
トライアル後半:	2024年4月1日 ~	2024年6月30日
対象工程:	コーディングフェーズ	
対象者:	初心者、中堅(計4名)	

2024年実施トライアル対象



[コーディングに生成AIを活用する理由]

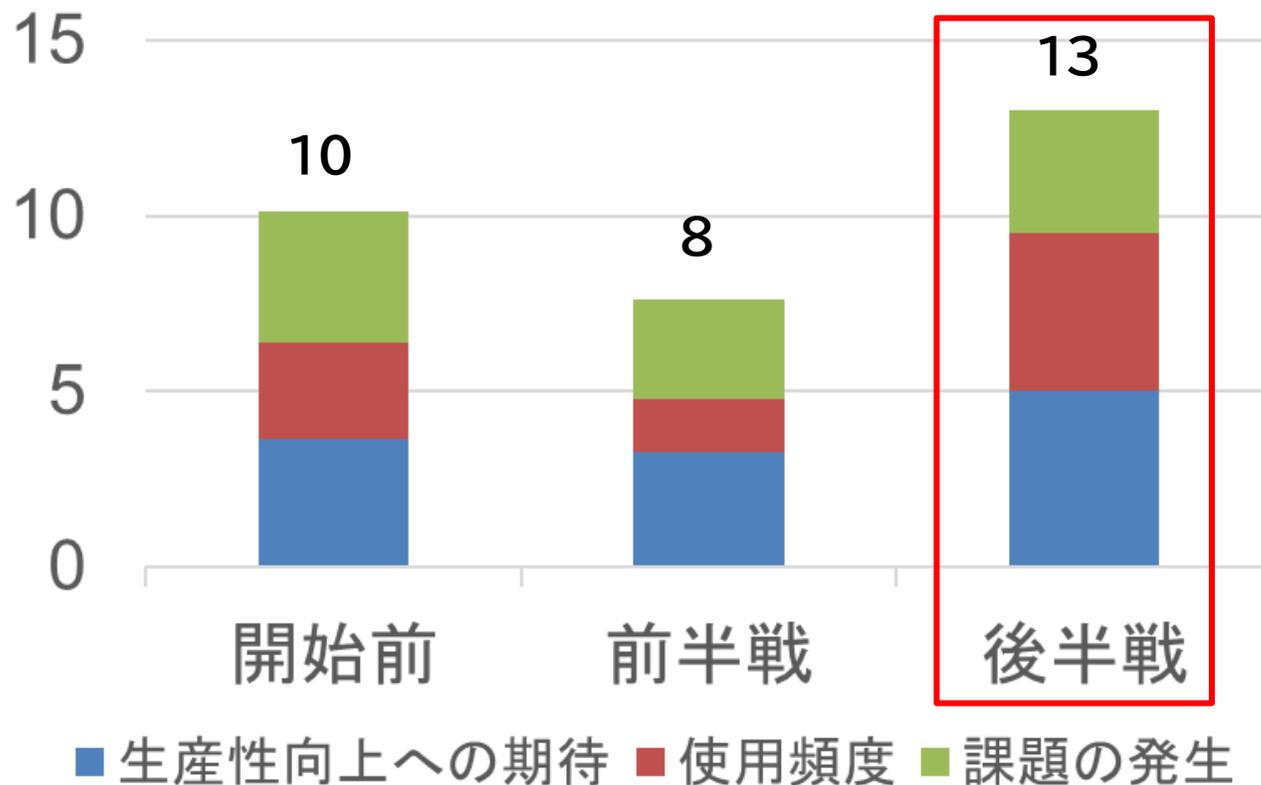
- 設計文書への生成AI活用は容易ではない。
- コード欠陥の混入の多くは単体テストで検出しやすく、失敗コストが低い。
- 若手エンジニアのプログラミング技術の早期獲得を図る。

2024年実施トライアル評価基準

評価項目	生産性判断基準
生産性	生産性向上が高まると回答した割合が75%以上である
使用実績	5割以上のケースで適用できた。
課題	業務上支障となる大きな課題が無かった。 課題はあっても対応可能なものであった。
再学習	生成AIを使用するために必要な学習が把握できている。
品質	人よりも品質の劣化が無い。

2024年実施トライアル結果

生成AIトライアルアンケート結果



各1-5点評価
・生産性向上への期待
・使用頻度
・課題の発生

※再学習、品質は対象外



前半戦では開始前より低下
後半戦で評価が高まった



後半で実施した教育が有効だった

2024年実施トライアルまとめ

昨年度の報告:

『生成AIは人材育成を前進させ

ソフトウェア開発の生産性向上に期待できる。』



結局、生成AIで生産性向上したの？



目的

生成AIによる実装工程の生産性向上を定量的に評価する。

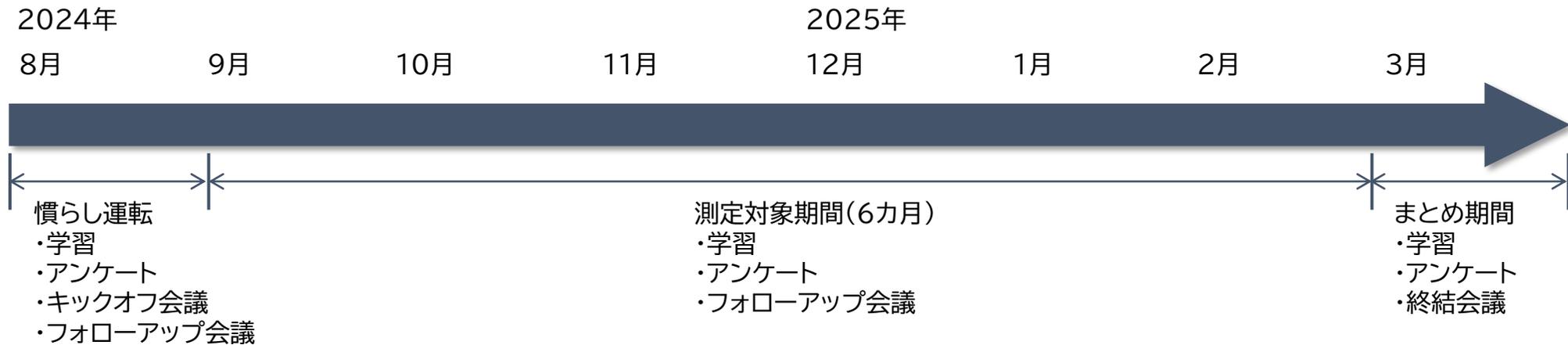
ソフトウェア生産性向上の可視化プロセス

既存のプロジェクト管理プロセスを活用した生産性可視化プロセスを確立し、AIエディターの導入効果を定量的に評価する。

これにより、開発プロセスへの影響を最小限に抑えながら、経営層への投資判断の根拠となる定量的なデータを提供する。

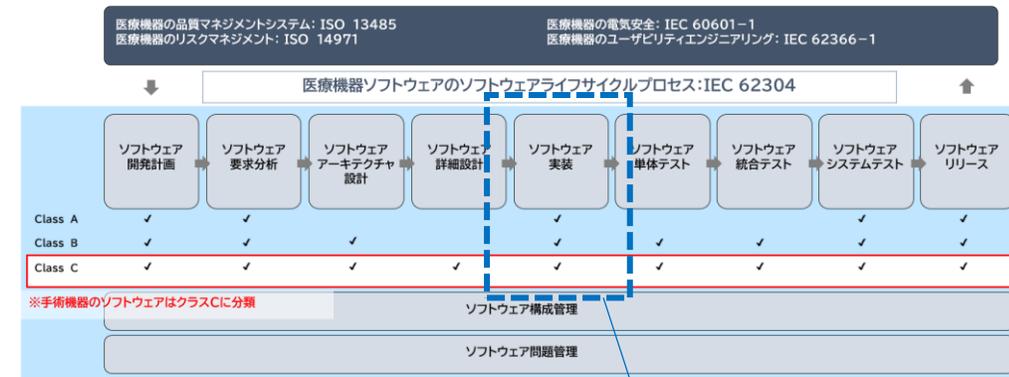


トライアル設計(2024-2025年の6ヵ月)



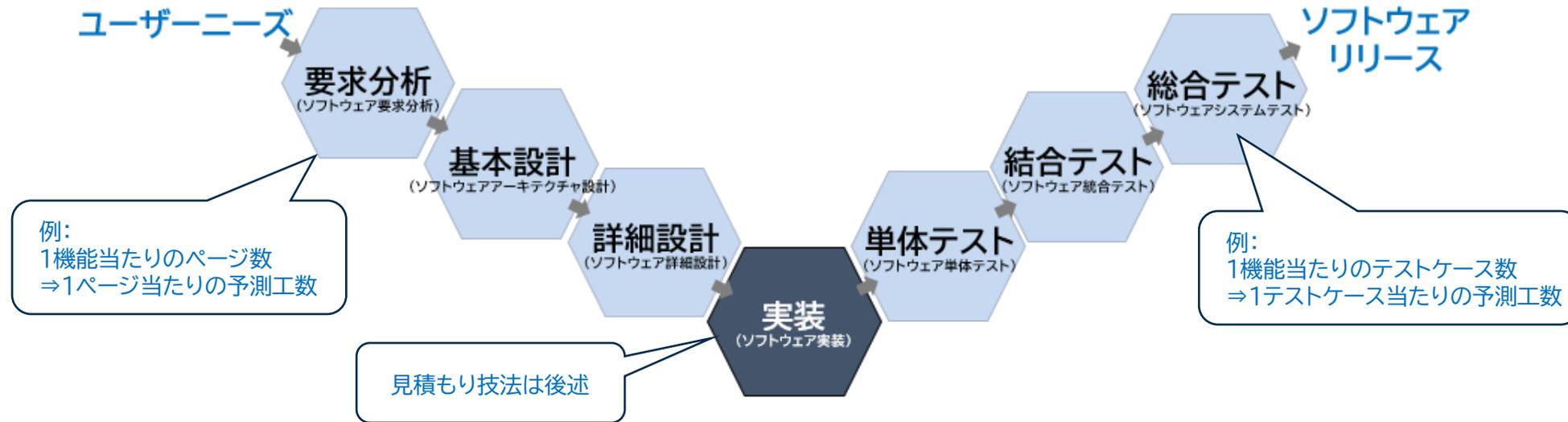
対象プロジェクト:7プロジェクト、13名
 (内訳)

- ・クラスA:2プロジェクト
- ・クラスB:3プロジェクト
- ・クラスC:2プロジェクト



トライアルの対象

プロジェクト管理プロセス:計画プロセス



生産性評価の対象は実装工程とする。

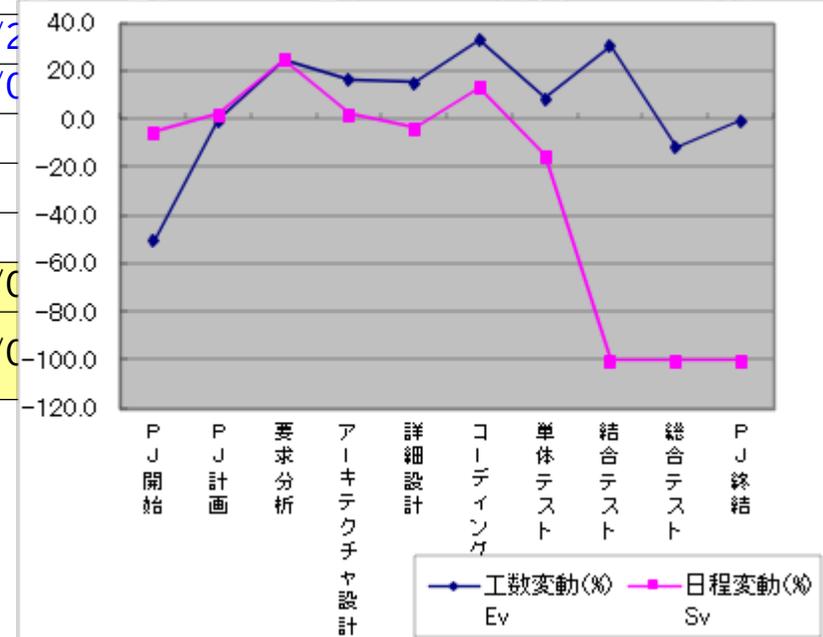


各フェーズごとの日程と資源(ヒト・モノ・カネ・情報)を計画する。
 プロジェクト管理プロセスは、共通フレーム(ISO/IEC 12207(JIS X0160))をベースとした建付けである。

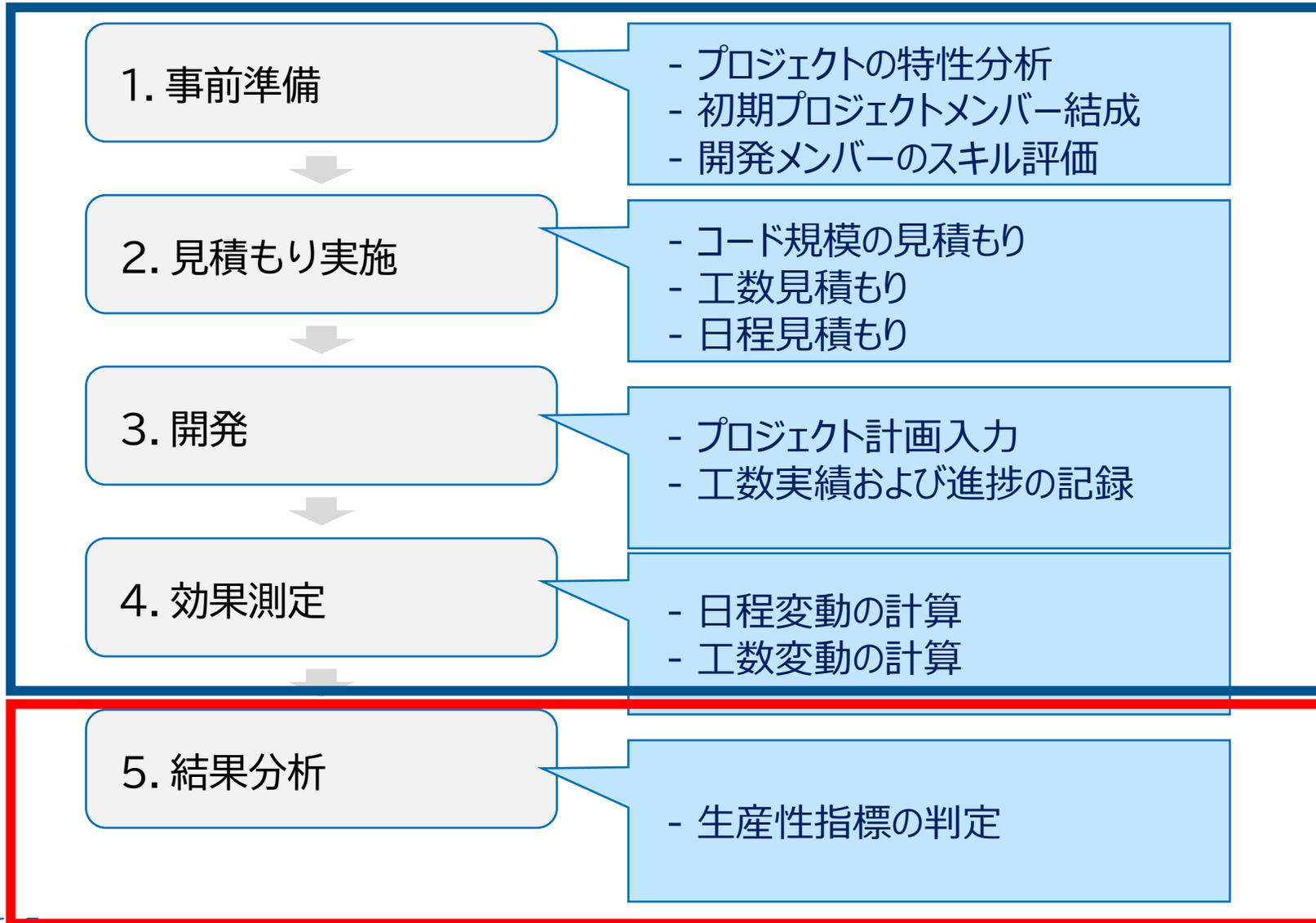
プロジェクト監視・制御

プロジェクトの監視および制御は管理ワークシートを使用する(後述)

フェーズ Phase	見積工数 (人時) Ep	実績工数 (人時) Ea	計画開始日	計画終了日	実績開始日	実績終了日	見積期間 (日数) DRp	実績期間 (日数) DRa	工数変動 (%) Ev	日程変動 (%) Sv
PJ開始	20	10	2017/04/01	2017/04/30	2017/04/01	2017/04/27	20	19	-50.0	-5.0
PJ計画	40	40	2017/05/01	2017/06/30	2017/04/28	2017/07/02	45	46	0.0	2.2
要求分析	240	300	2017/07/01	2017/08/31	2017/07/03	2017/09/15	44	55	25.0	25.0
アーキテクチャ設計	360	420	2017/09/01	2017/10/30	2017/09/16	2017/11/15	42	43	16.7	2.4
詳細設計	520	600	2017/11/01	2017/12/10	2017/11/16	2017/12/22	28	27	15.4	-3.6
コーディング	360	480	2017/12/11	2018/01/30	2017/12/23	2018/02/22				
単体テスト	440	480	2017/12/21	2018/02/25	2018/01/07	2018/03/06				
結合テスト	520	680	2017/11/15	2018/03/31	2017/12/01					
総合テスト	360	320	2017/10/01	2018/04/30	2017/10/15					
PJ終結	15	15	2018/05/01	2018/05/20						
合計	2875	3345	2017/04/01	2018/05/20	2017/04/01	2018/03/06				
合計(実績進捗率100%のフェーズのみ)	60	50	2017/04/01	2017/06/30	2017/04/01	2017/07/02				



可視化プロセスの全体像



既存のプロジェクト管理プロセス

可視化のための追加プロセス



評価指標の選定

1. 工数変動 : Ev

- $Ev = (Ea - Ep) / Ep * 100$
- Ep : **見積もり工数**
- Ea : **実績工数**
- 工数変動は計画に対する実績工数の増減を表す

2. 日程変動 : Sv

- $Sv = (DRa - DRp) / DRp * 100$
- DRp : **見積もり期間**
- DRa : **実績期間**
- 日程変動は計画に対する実績の遅れ/早まりを表す

フェーズ Phase	見積工数 (人時) Ep	実績工数 (人時) Ea	計画開始日	計画終了日	実績開始日	実績終了日	見積期間 (日数) DRp	実績期間 (日数) DRa	工数変動 (%) Ev	日程変動 (%) Sv
PJ開始	20	10	2017/04/01	2017/04/30	2017/04/01	2017/04/27	20	19	-50.0	-5.0
PJ計画	40	40	2017/05/01	2017/06/30	2017/04/28	2017/07/02	45	46	0.0	2.2
要求分析	240	300	2017/07/01	2017/08/31	2017/07/03	2017/09/15	44	55	25.0	25.0
アーキテクチャ設計	360	420	2017/09/01	2017/10/30	2017/09/16	2017/11/15	42	43	16.7	2.4
詳細設計	520	600	2017/11/01	2017/12/10	2017/11/16	2017/12/22	28	27	15.4	-3.6
コーディング	360	480	2017/12/11	2018/01/30	2017/12/23	2018/02/20	37	42	33.3	13.5
単体テスト	440	480	2017/12/21	2018/02/25	2018/01/07	2018/03/02	47	40	9.1	-14.9
結合テスト	520	680	2017/11/15	2018/03/31	2017/12/01		98	0	30.8	-100.0
総合テスト	360	320	2017/10/01	2018/04/30	2017/10/15		151	0	-11.1	-100.0
PJ終結	15	15	2018/05/01	2018/05/20			14	0	0.0	-100.0
合計	2875	3345	2017/04/01	2018/05/20	2017/04/01	2018/03/02	295	240	16.3	-18.6
合計(実績進捗率100%の フェーズのみ)	60	50	2017/04/01	2017/06/30	2017/04/01	2017/07/02	65	65	-16.7	0.0

生産性基準値の設定

AIエディター導入にあたって効果判定基準として日程変動を最低5%の改善を基準として設定した。

(スクラムの判定については後述)

改善策の内容

トライアル対象プロジェクトの拡大。拡大における可視化プロセスの標準化

1. 評価対象の拡大

- 生成AIによるプログラミングに不慣れなメンバーも含めて評価対象とした

2. 評価基準の統一

- 当社の標準作業手順（SOP）に従ってプロジェクトの見積もりを実施 ⇒ プロジェクトデータを活用できる
- 見積もりには、コード規模、工数、日程を含む

3. アジャイル開発での生産性評価

スクラムチームでは、見積もり技法が異なるため、ベロシティ推移による比較を行うこととした。

可視化プロセスの実施手順

表 1 ウォーターフォール型開発とアジャイル型開発の実施手順比較

ステップ	ウォーターフォール型開発	アジャイル型開発
1. 事前準備	<ul style="list-style-type: none">- プロジェクトの特性分析- 初期プロジェクトメンバー結成- 開発メンバーのスキル評価	<ul style="list-style-type: none">- プロジェクトの特性分析- 初期プロジェクトメンバー結成- 開発メンバーのスキル評価
2. 見積もり実施	<ul style="list-style-type: none">- コード規模の見積もり- 工数見積もり- 日程見積もり(*1)	<ul style="list-style-type: none">- ストーリーポイントによる見積もり- スプリントバックログの作成- チームベロシティの設定
3. 開発実施	<ul style="list-style-type: none">- プロジェクト計画入力- 工数実績および進捗の記録	<ul style="list-style-type: none">- スプリントごとの完了ストーリーポイントの計測- バックログの実績記録
4. 効果測定	<ul style="list-style-type: none">- 日程変動の計算	<ul style="list-style-type: none">- スプリントごとのベロシティ計算
5. 結果分析	<ul style="list-style-type: none">- プロジェクトごとの日程変動の評価- 効果とプロジェクト特性との関係の評価	<ul style="list-style-type: none">- 導入前後のベロシティの評価- 効果とプロジェクト特性との関係の評価

使用するAIEDィターの特徴

1. 主な機能

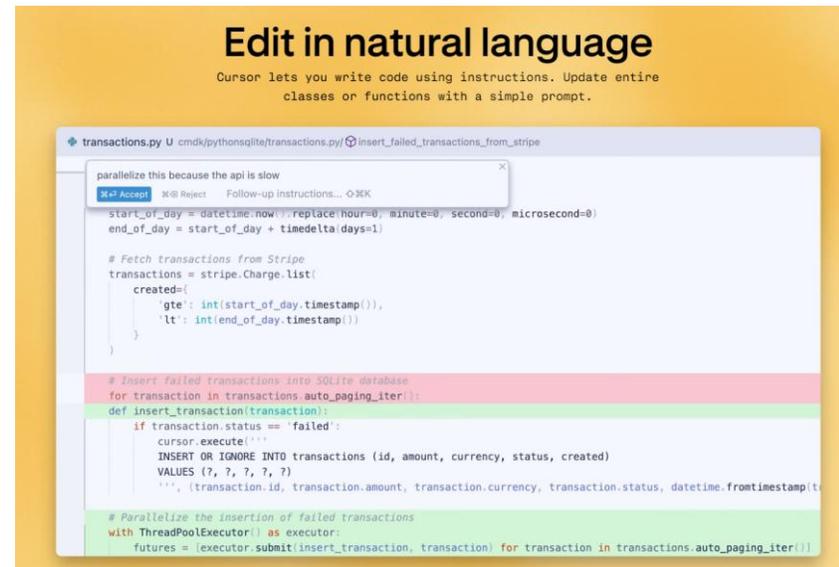
- コード補完・生成機能
- コードレビュー支援
- バグ修正提案
- ドキュメント生成支援
- プロジェクト固有の設定管理
- チーム間での設定共有

2. 選定理由

- 高いコード生成精度
- 日本語でのプロンプト入力に対応
- セキュリティ要件を満たす
- 既存の開発環境との親和性が高い

3. 使用上の制約と対策

- セキュリティ要件
- 機密情報の取り扱いルールの確立
- 品質要件
- 生成コードのレビュー基準の確立
- テスト自動化との連携



見積もりプロセス

No.	機能	規模 (LOC)	複雑度 (表1)	生産性 (LOC/人月)	見積工数(人月)
1	例:アプリケーション	7833	高	2433	3.22
2	例:送液管理	1517	中	3000	0.51
3	例:電源制御	2300	低	4000	0.58
4	例:センサ監視	900	低	4000	0.23
5	例:UI入力	900	中	3000	0.30
6	例:UI出力	12033	高	2433	4.95
7	例:警報・ログ管理	2783	中	3000	0.93
8	例:通信	3783	低	4000	0.95
9	例:システム制御・OS	1517	中	3000	0.51
10	例:データ管理	7906	中	3000	2.64
	合計	41473	-	2805	14.78

三点見積り法を適用



複雑度	LOCに換算した生産性(LOC/人月)			
	楽観の見積 (A)	悲観の見積 (B)	最もありそうな見積(M)	予想値 = (A+4*M+B)/6
高	3000	1800	2500	2433
中	3500	2500	3000	3000
低	4500	3500	4000	4000

データ収集と評価プロセス(再掲)

1. 工数変動 : Ev

- $Ev = (Ea - Ep) / Ep * 100$
- Ep : **見積もり工数**
- Ea : **実績工数**
- 工数変動は計画に対する実績工数の増減を表す
- 実績工数はTimeTrackerを使用して集積する

2. 日程変動 : Sv

- $Sv = (DRa - DRp) / DRp * 100$
- DRp : **見積もり期間**
- DRa : **実績期間**
- 日程変動は計画に対する実績の遅れ/早まりを表す

フェーズ Phase	見積工数 (人時) Ep	実績工数 (人時) Ea	計画開始日	計画終了日	実績開始日	実績終了日	見積期間 (日数) DRp	実績期間 (日数) DRa	工数変動 (%) Ev	日程変動 (%) Sv
PJ開始	20	10	2017/04/01	2017/04/30	2017/04/01	2017/04/25	20	19	-50.0	-5.0
PJ計画	40	40	2017/05/01	2017/06/30	2017/04/28	2017/07/02	45	46	0.0	2.2
要求分析	240	300	2017/07/01	2017/08/31	2017/07/03	2017/09/15	44	55	25.0	25.0
アーキテクチャ設計	360	420	2017/09/01	2017/10/30	2017/09/16	2017/11/15	42	43	16.7	2.4
詳細設計	520	600	2017/11/01	2017/12/10	2017/11/16	2017/12/22	28	27	15.4	-3.6
コーディング	360	480	2017/12/11	2018/01/30	2017/12/23	2018/02/20	37	42	33.3	13.5
単体テスト	440	480	2017/12/21	2018/02/25	2018/01/07	2018/03/02	47	40	9.1	-14.9
結合テスト	520	680	2017/11/15	2018/03/31	2017/12/01		98	0	30.8	-100.0
総合テスト	360	320	2017/10/01	2018/04/30	2017/10/15		151	0	-11.1	-100.0
PJ終結	15	15	2018/05/01	2018/05/20			14	0	0.0	-100.0
合計	2875	3345	2017/04/01	2018/05/20	2017/04/01	2018/03/02	295	240	16.3	-18.6
合計(実績進捗率100%のフェーズのみ)	60	50	2017/04/01	2017/06/30	2017/04/01	2017/07/02	65	65	-16.7	0.0

アジャイル開発における評価方法

1. スプリントバーンダウンチャート

- 日次のタスク完了状況を可視化
- AIEディター使用によるタスク完了速度の変化を測定
- チーム全体の生産性トレンドを把握

2. ベロシティの測定

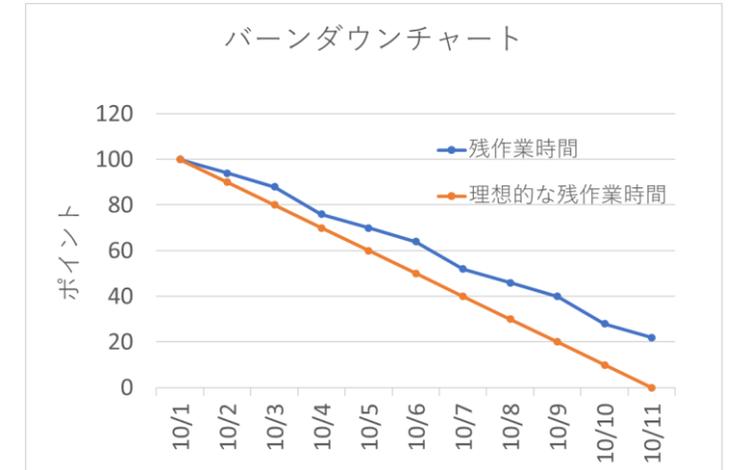
- スプリントごとの完了ストーリーポイントを計測
- AIEディター導入前後のベロシティを比較
- チームの生産性向上を定量的に評価

3. ベロシティ向上率の計算

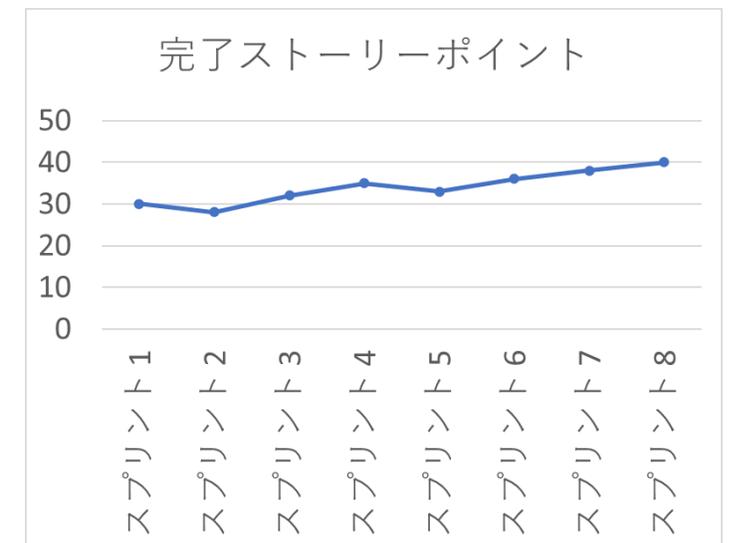
- ベロシティ向上率

$$= (\text{導入後ベロシティ} - \text{導入前ベロシティ}) / \text{導入前ベロシティ} \times 100$$

(参考データ)

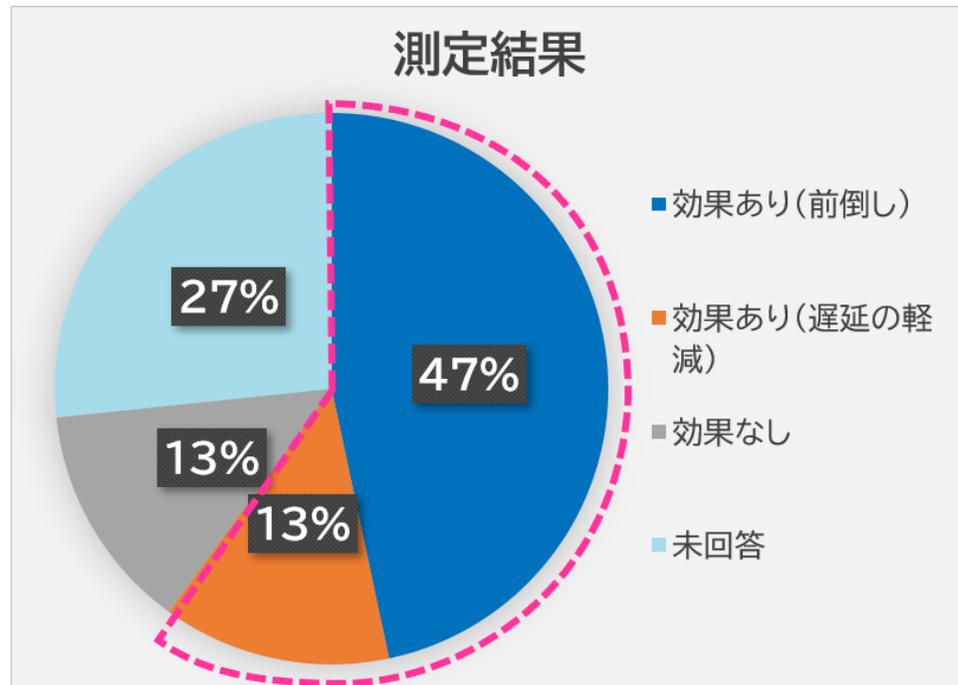


(参考データ)



改善による変化や効果

AIエディター導入による生産性向上効果の測定結果を示す。**60%のプロジェクトで効果**が確認され、効果があったプロジェクトにおいては、**20%**（中央値）の生産性向上が確認された。



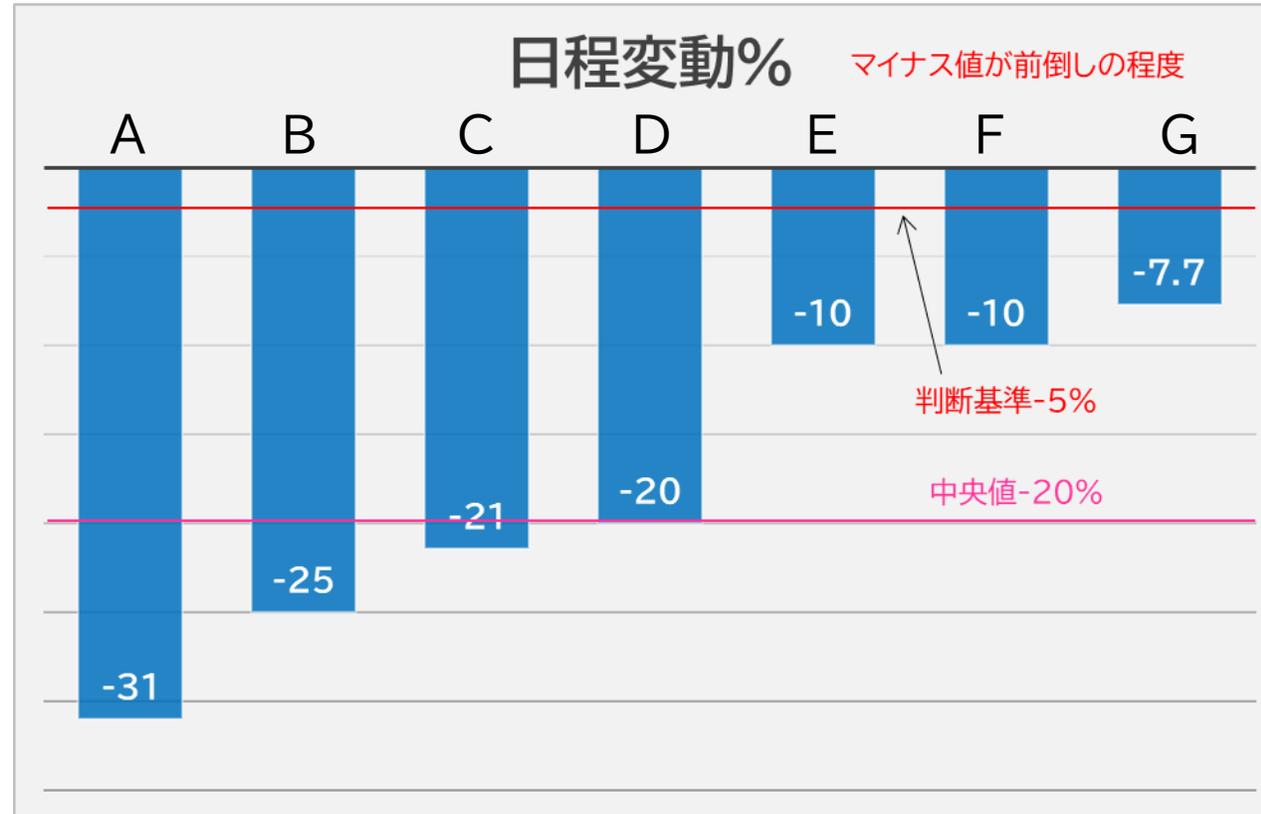
効果が高かったプロジェクトの特徴

- 新規開発
- 仕様が明確
- 宣言的記述でGUI設計
- コードが難解
- スクラム開発

効果が無かったプロジェクトの特徴

- 初心者
- コーディング以外の計画外作業の多いプロジェクト
- プロジェクト管理者
- レガシーコードの保守
- 独自のコーディング規約（対応により遅れが発生）

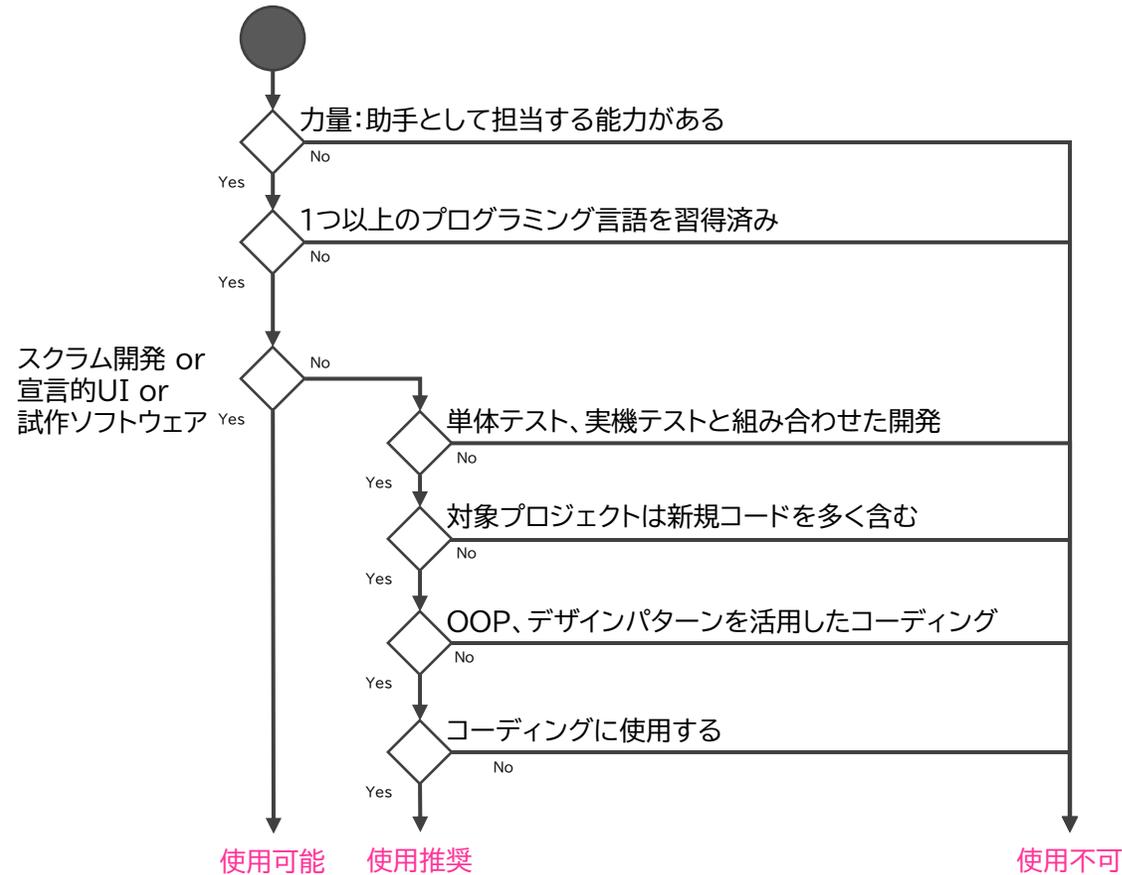
プロジェクトごとの日程変動の結果



効果が高かったプロジェクトの特徴

- 新規開発
- 仕様が明確
- 宣言的記述でGUI設計
- コードが難解

生成AI利用基準



力量レベル

- ・A:完全に理解し指導する能力がある
- ・B:一人で担当する能力がある
- ・C:助手として担当する能力がある
- ・D:指導されて担当する能力がある
- ・E:担当する能力がない

まとめ

既存のプロジェクト管理プロセスをしようすることで、生成AIの効果を定量的に評価できる。

プロジェクト管理によるプロジェクトの可視化は、現場の健全化だけでなく、経営判断にも資するため、経営層がプロジェクトの生産性に関心を持ち、さらなる改善につながる。

そのためには、生産性向上という目標をスローガンではなく、定量的な指標で評価・共有することが重要である。



参考文献

- [1] 山崎裕司（株式会社ニデック）, 「AI共創開発：新時代の生産性と人材育成に関するトライアル～生成AI活用によるソフトウェア開発プロセスと人材育成の最適化～」, SPI Japan 2024
- [2] IEC 62304, "Medical device software - Software life cycle processes", International Electrotechnical Commission, 2006
- [3] スティーブ・マコネル著、石井信明訳『ソフトウェア見積り』日経BPソフトプレス、2007年、ISBN978-4-89100-467-1
- [4] デンソークリエイト株式会社, "TimeTracker - 工数管理システム", <https://www.denso-create.co.jp/products/timetracker/>, 2024
- [5] Anysphere Inc., "Cursor - AI-first code editor", <https://cursor.sh/>, 2024

ご清聴ありがとうございました