AI駆動開発の品質管理ガイドライン

生成AIを活用した開発・レビュー・テストの
品質を担保する実践手法

2026年8月　第1版



Start Challenge Consulting co., ltd.
挑む力が、世界を照らす

Confidential㊙社外秘

はじめに

生成AIの活用によって開発速度は向上するが、成果を左右するのは「生成物をどう検証し、誰が責任を持って承認するか」であり、
本資料はAI生成物を前提とした品質管理の実務手順を、工程・体制・指標・証跡の4側面から定義する。

背景｜AI駆動開発の広がり

・要件整理から設計・コード・テストまで、開発工程の広い範囲で生成AIの利用が前提化しつつある

・個人の補助ツールとしての利用から、**チーム・組織の標準プロセスへの組み込み**へと段階が移行

・AIエージェントが複数工程を連続実行する構成では、人の関与点が曖昧になりやすい

論点｜速度向上だけでは成功しない

・生成の速さは、検証・修正・手戻りの工数を含めた**総所要時間**で評価されなければ意味を持たない

・仕様の誤解、保守しにくい構造、脆弱性の混入は、リリース後に費用として顕在化する

・「動いた」ことと「品質基準を満たした」ことは同義ではない

目的｜本資料が定義するもの

・AI生成物を「レビュー対象」として扱うための**工程別の品質管理手順**

・生成者と承認者を分離する体制・責任分担（RACI）

・品質ゲート、評価指標、ログ・証跡の管理要件

・段階的な導入ロードマップと定着のための確認項目

本資料の構成

Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6
背景と全体像 P2–P4	リスクと管理方針 P5–P7	工程別の品質管理 P8–P12	体制とゲート設計 P13–P15	測定と証跡 P16–P17	導入と定着 P18–P20

対象読者
Target Audience

経営者 ／ PM・PMO ／ 開発責任者 ／ エンジニア ／ 品質管理担当者 ／ セキュリティ担当者

＊立場により参照範囲が異なる：経営者・PMOはP6・P7・P15・P16／開発・品質担当はP8～P12・P17を主に参照

注：本ページは本資料の位置づけを示す。事実：AI生成物は工程間で連鎖する／解釈：検証点の不在が手戻りの主要因となる／提案：工程ごとに品質ゲートを設定し、人の承認を必須とする

AI駆動開発とは

AI駆動開発は「人の作業をAIが補助する」ものではなく、**生成AIを標準工程として組み込み、人は判断と承認に集中する**開発形態である。
したがってAIの担当範囲が広がるほど、成果物の検証責任と最終責任の所在を明示することが前提条件となる。

1.兼I支援開発との違い

観点	AI支援開発	AI駆動開発
位置づけ	個人の生産性向上ツール	**開発プロセスの標準構成要素**
利用範囲	実装の一部（補完・雛形生成）	**要件整理～テスト・文書化まで**
品質管理	既存のレビュー手順に依存	**AI生成物専用のゲートを追加**
主なリスク	局所的な品質ばらつき	**誤りが工程間で連鎖・増幅する**

2.兼Iが担当できる主な作業

01 **要件整理** 論点抽出・不明点リスト化	02 **設計書作成** 設計案の複数提示・比較	03 **コード生成** 実装・差分修正・リファクタ
04 **コードレビュー** 規約適合・欠陥候補の指摘	05 **テストケース生成** 異常系・境界値の網羅補強	06 **ドキュメント作成** 手順書・仕様書・引継資料

基本原則｜人が最終責任を持つ

①兼責任は委譲できない
AIは作業の実行主体になり得るが、責任主体にはならない。生成物の妥当性判断と承認は、常に指名された人が行う。

②兼判断はAIに委ねない
業務要件の確定、方式選定、リスク受容の判断は人の領域。AIは選択肢と根拠の提示までを担当する。

③兼検証可能な状態を保つ
誰が・どの入力で・何を生成し・どこを修正して承認したかを記録し、後から説明できる状態を維持する。

注：AIの担当範囲は組織の成熟度により異なる。事実：AIは6領域の作業を生成可能／解釈：担当範囲の拡大は誤りの連鎖リスクを増加させる／提案：範囲拡大と同時に承認点を定義する

AI駆動開発の全体プロセス

AI生成物を検証せずに次工程の入力とすると、誤りが下流工程で増幅し、テスト・リリース段階での手戻りに転化する。
そのため各工程の出口に**品質ゲート（G1〜G5）を設置し、確認者を特定した状態で次工程へ引き渡す**ことを標準とする。

	① 要件定義	② 設計	③ 実装	④ テスト	⑤ リリース・運用
入力情報 人が用意する	業務要件・現行仕様 制約条件・非機能要件	確定要件・受入条件 既存アーキテクチャ	承認済み設計書 コーディング規約	仕様・受入条件 レビュー済みコード	変更内容・影響範囲 リスク評価結果
AIの生成物 レビュー対象	要件整理案 不明点・前提の一覧	設計案（複数） 方式比較表・構成図案	コード・修正差分 実装メモ	テストケース テストコード・結果整理	リリース手順書 運用手順・障害対応案
人の役割・確認者 承認の主体	**業務・仕様責任者** 要件の確定と受入条件の合意	**設計責任者** 方式選定とリスク受容の判断	**レビュー担当（生成者と別）** 規約・構造・脆弱性の確認	**品質管理担当** 仕様との対応と網羅性の判定	**最終承認者** リリース可否と監視体制の確認
品質ゲート	G1 要件品質ゲート	G2 設計品質ゲート	G3 コード品質ゲート	G4 テスト品質ゲート	G5 リリース品質ゲート

禁止事項｜AI生成物を無検証で次工程の入力にしない

未確認の要件整理案から設計を生成し、その設計からコードとテストを連続生成すると、初期の誤解が全工程に伝播したうえ、**テスト自体が誤った仕様を正解として固定する**。ゲート未通過の生成物は、次工程の入力として使用しない。

横断ゲート｜セキュリティ品質（GS）

認証・認可、個人情報、外部連携、本番データを扱う変更は、③〜⑤を横断してセキュリティ担当の確認を必須とする（詳細はP11・P15）。

注：工程名・ゲート名称は本資料の標準定義。事実：生成物は工程間で入力として再利用される／解釈：検証点のない引き渡しが誤りを増幅させる／提案：G1〜G5＋GSを標準ゲートとする

AI駆動開発で発生する品質リスク

AI駆動開発のリスクは実装品質にとどまらず、**要件の誤解が上流で発生し、AI生成物同士の相互参照によって誤りが正当化される構造**にある。
したがって発見が遅れるほど影響が拡大する7類型を、検知工程とともに定義して管理する必要がある。

R1｜要件の誤解釈

曖昧な指示を前提付きで補完し、業務上あり得ない仕様として確定させる。
検知：G1／業務・仕様責任者

R2｜ハルシネーション

存在しないAPI・引数・設定項目を、実在するかのような記述で生成する。
検知：G2・G3／ビルド・静的解析

R3｜保守困難なコード

動作するが責務が混在し、重複と暗黙依存を含む構造となる。
検知：G3／レビュー担当

R4｜古い・非推奨の実装

学習時点の情報に基づき、非推奨機能や旧バージョン前提の実装を出力する。
検知：G3／依存関係チェック

R5｜セキュリティ脆弱性の混入

入力値検証の欠落、認証情報のハードコード、権限過剰付与などが、外形上正常なコードとして残る。
検知：GS／セキュリティ担当

R6｜テスト不足・異常系の見落とし

正常系に偏り、境界値・権限別・障害時の挙動が試験されないまま網羅率だけが上がる。
検知：G4／品質管理担当

R7｜AI生成物同士による誤りの正当化

同一モデル・同一前提でコードとレビューとテストを生成すると、同じ誤解が相互に肯定され、欠陥が「合格」として通過する。
検知：G3・G4／別基準のレビュー＋人の承認

発見が遅れた場合の影響の深化

発生事象	直接原因	構造的原因	業務への影響	経営への影響
AI生成コードが本番で想定外の挙動を示す（データ不整合・処理停止・情報漏えい）	異常系・境界値の試験が不足し、レビューが生成AIの出力を追認しただけで終わっていた	生成者と承認者が同一で、品質ゲートと合否基準が定義されていない（R7が機能を継続）	障害対応と再作業で開発工数が増加し、AI活用による短縮効果が相殺される	**顧客・監督官庁への説明責任を果たせず、AI活用方針そのものの信頼性が損なわれる**

注：リスク番号R1～R7は本資料の整理。事実：生成AIは学習範囲外の情報を断定的に出力し得る／解釈：同一モデルでの自己検証は独立性を欠く／提案：検知工程と責任者をリスク別に割り当てる

品質管理の基本方針

AI生成物の品質は「入念に見る」という運用努力では安定しない。**確認レベルの配分、生成者と承認者の分離、指標による管理**を含む6原則を規程として定め、属人的な判断に依存しない運用へ移行する。

原則1｜AIの出力は「レビュー対象」

生成物は成果物候補であり、承認を経て初めて成果物となる。「AIが生成したから」はレビュー省略の理由にならない。

運用：生成物は必ず差分レビューを経てマージする

原則2｜重要度に応じた確認レベル

全件を同一水準で見ると確認が形式化する。影響範囲・可逆性・データ機密性でリスク区分し、確認の深さを変える。

運用：高／中／低の3区分と必須確認者を定義（下表）

原則3｜入力・出力・修正履歴を記録

モデル・プロンプト・参照資料・生成物・修正内容・承認者を記録し、後から再現・説明できる状態を保つ。

運用：記録項目はP17で定義し、変更管理と紐付ける

原則4｜生成者と承認者の分離

生成を担当した人・AIが自らの成果を承認しない。レビューAIは開発AIと別の指示・評価基準で運用する。

運用：体制とRACIをP7、エージェント構成をP13で定義

原則5｜感覚ではなく指標で管理

「速くなった」だけでは効果を評価できない。修正率・不具合検出率・手戻り工数を継続測定し、速度と品質を併せて判断する。

運用：評価指標をP16で定義し、月次で推移を確認

原則6｜人が確認する範囲を事前定義

介在点を事後判断に委ねず、要件確定・設計決定・本番データ・セキュリティ・リリースの5点を必須とする。

運用：Human-in-the-Loop設計をP14で定義

原則2の具体化｜リスク区分別の確認レベル

リスク区分	該当する変更の例	必須の確認行為	承認者
高	認証・認可、決済、個人情報、本番データ移行、外部公開API	人による全行レビュー＋セキュリティレビュー＋異常系試験	**最終承認者＋セキュリティ担当（2名以上）**
中	業務ロジック変更、画面・帳票の仕様変更、内部連携の追加	人による差分レビュー＋レビューAIの指摘対応＋回帰テスト	**レビュー担当（生成者以外）**
低	文言修正、整形、コメント追記、内部ドキュメントの更新	静的解析・自動テスト通過を条件に差分の要点確認	**開発チーム内レビュー（1名）**

注：リスク区分の該当例・承認者数は導入時に組織の規程へ合わせて調整する。事実：確認負荷は変更件数に比例する／解釈：一律確認は形式化を招く／提案：3区分で確認の深さを配分する

品質管理体制と責任分担

AIエージェントを利用しても、責任は**そのエージェントを起動した担当者と、その工程の承認者に帰属する。**
したがって役割を6つに分解し、RACIによって「生成する人」と「承認する人」が同一にならない状態を構造として担保する。

プロジェクト責任者

AI利用方針・品質基準の決定、ゲート運用の維持、例外リリースの判断。品質と進捗の最終説明責任を負う。

AI開発担当者

プロンプト設計と生成の実行、生成物の一次確認と修正、入力・出力・修正履歴の記録。**自らの生成物を承認しない。**

AIレビュー担当者

開発側とは別の評価基準でレビューを実施。レビューAIの指摘の妥当性判断と、指摘漏れの補完を行う。

セキュリティ担当者

AIへの入力制限の設計と監視、脆弱性・権限設計の確認、外部AIサービスへのデータ送信範囲の承認。

業務・仕様責任者

要件・受入条件の確定、AIが推測した前提の是非判断、業務影響の評価。上流の誤解を止める役割。

最終承認者

ゲート通過状況と残存リスクを確認し、リリース可否を判断。判断根拠と記録の保全を確認する。

RACIによる役割整理例

R＝実行　A＝説明責任（承認）　C＝相談　I＝報告受領　／　網掛けは同一人物での兼務を禁止する組合せ

工程・活動	プロジェクト責任者	AI開発担当者	AIレビュー担当者	セキュリティ担当者	業務・仕様責任者	最終承認者
要件の確定（G1）	C	R	C	I	**A**	I
設計方式の決定（G2）	**A**	R	C	C	C	I
コード生成・修正	I	**R**	C	I	I	I
コードレビュー承認（G3）	I	兼務禁止	**A**	C	I	I
セキュリティ確認（GS）	I	R	C	**A**	I	C
リリース判定（G5）	C	I	C	C	C	**A**

注：RACIの割当ては例であり、組織規模に応じて兼務の可否を定める。事実：AIエージェントは責任主体になり得ない／解釈：起動者と承認者への帰属が必要／提案：Aは各工程1名に限定する

要件定義工程の品質管理

曖昧な要件をAIへ渡すと、AIは不足部分を推測で補完し、その推測が確定仕様として下流に流れる。
したがって**「確定事項」と「AIの推測」をラベルで分離し、受入条件を検証可能な形で定義したうえでG1を通過させる**ことが要件工程の管理点となる。

1. AIへ渡す前の要件整備

避けるべき入力（曖昧な指示）	整備後の入力（検証可能な指示）
・「使いやすい申請画面をつくる」 ・「エラーは適切に処理する」 ・「処理は十分に速くする」 ・「既存仕様に合わせる」 → AIが対象範囲・閾値・例外を推測で補完	・対象業務・利用者・件数を明記 ・異常時の表示内容と後続処理を定義 ・応答時間の上限値と測定条件を明記 ・参照すべき既存仕様書の版を指定 → 推測の余地を減らし、検証可能にする

2. 要件の3分類と記述の必須項目

機能要件	非機能要件	制約条件
・利用者と操作の単位 ・入力・出力・保存項目 ・業務ルールと例外処理 ・権限別に許可される操作	・性能（応答時間・同時件数） ・可用性と復旧目標 ・セキュリティ要求水準 ・保守性・監視・ログ要件	・利用可能な技術・製品 ・社内規程・法令要求 ・既存システムとの互換性 ・AIへの入力禁止データ

3. 確定事項とAIの推測を区別する記述ルール（要件一覧への必須列）

区分ラベル	意味	受入条件の記述例	G1通過の条件
【確定】	業務・仕様責任者が合意した事項	申請金額が上限を超える場合、登録を拒否し理由コードE102を返す	**合意記録があり、テスト可能な条件で記述されている**
【AI推測】	AIが不足情報を補完した内容	上限額は100万円と想定（根拠：類似機能の既定値）	**責任者の確認により【確定】へ変更されるまで実装対象外**
【未決】	論点として残っている事項	権限グループ別の承認段数（業務部門と調整中）	**決着期限と担当が明記されている（未決のままG2へ進めない）**

4. 要件と生成物の追跡（トレーサビリティ）

要件IDを設計書・コミット・テストケースに付与し、「どの要件がどのコードとテストで実現されたか」を双方向に辿れる状態を維持する。

5. 業務責任者による確認ポイント

①【AI推測】の全件を可否判断　②例外・異常時の業務手順の妥当性　③権限別の操作範囲　④受入条件がテストで確認できる記述か。

注：金額・コード等は記述形式を示す例示。事実：AIは不足情報を推測で補完する／解釈：推測が無標記のまま確定仕様化する／提案：3区分ラベルを要件一覧の必須列とする

設計工程の品質管理

設計工程では、AIに**単一の設計案ではなく複数案と評価軸を生成させ、選定と方式決定は人が行う。**
特に認証・認可、エラー処理、ログ・監視は生成物が形式的に整っていても要件との不整合が残りやすく、G2の重点確認対象となる。

1. 設計レビューの5領域とチェック項目

領域	確認内容	確認者
アーキテクチャ	既存構成との整合、責務分割の妥当性、性能・拡張要件への適合、採用技術の保守性	プロジェクト責任者／設計責任者
データフロー・外部連携	データの発生・保持・削除の経路、連携先の仕様と実在確認、整合性の担保方式、送信されるデータ項目の範囲	設計責任者／業務・仕様責任者
認証・認可・権限設計	認証方式と有効期間、権限モデルと最小権限、権限昇格の経路、代理操作・管理者操作の記録	セキュリティ担当（必須）
エラー処理・リトライ	異常系の分類と業務手順、リトライ回数・間隔・上限、二重処理の防止、部分失敗時の整合回復	設計責任者／業務・仕様責任者
ログ・監視・バックアップ	記録項目と保存期間、機密情報の出力禁止、検知条件と通知先、復旧目標と復旧手順の実施可能性	設計責任者／運用担当

2. AI生成設計案の比較評価（記述様式）

評価軸	案A	案B
要件充足	全要件を充足	一部を将来対応
既存構成との整合	整合	基盤変更が必要
運用・保守負荷	中	低
セキュリティ	既存基盤を流用	新規審査が必要
残存リスク	性能上限の検証要	移行期間の並行運用
人による選定（記録必須）	**採用** **選定理由を記録**	不採用 不採用理由を記録

3. G2（設計品質ゲート）通過条件

- ☐ 全ての要件IDが設計要素に割り当てられている
- ☐ 設計案が2案以上比較され、選定理由が記録されている
- ☐ 外部連携先のAPI・仕様が実在確認済み（ハルシネーション排除）
- ☐ 認証・認可・権限設計がセキュリティ担当の確認済み
- ☐ 異常系・リトライ・二重処理防止の方針が定義済み
- ☐ ログ出力項目に機密情報が含まれないことを確認済み
- ☐ 監視の検知条件・通知先・復旧手順が実施可能
- ☐ 【未決】要件が設計前提に含まれていない
- ☐ 残存リスクと受容判断が記録されている

注：比較表の案A・案Bは記述様式を示す例示。事実：AIは単一案を確定的な記述で出力しやすい／解釈：単一案の提示は選定の検討過程を欠く／提案：2案以上の比較と選定理由の記録を必須とする

AI生成コードの品質管理

AI生成コードは動作要件を満たしていても、責務の混在・重複・過剰な依存追加によって**保守コストを将来へ繰り延べる**形になりやすい。
そのためG3では動作確認に加え、可読性・構造・依存関係を6観点で判定し、生成元がAIであることをレビュー省略の理由としない。

#	観点	確認内容	AI生成物で頻出する不備	主な検出手段
1	**可読性・保守性**	命名の一貫性、関数長・分岐の深さ、意図が読み取れるコメント	長大な関数に処理が集約され、変更影響が読めない	静的解析＋人のレビュー
2	**規約適合**	コーディング規約・書式・ディレクトリ構成・ログ出力方針への適合	一般的な流儀で書かれ、社内規約と乖離する	Lint／フォーマッタの自動適用
3	**モジュール分割と責務**	層構造の遵守、業務ロジックと入出力の分離、循環依存の不在	画面処理内に業務ルールとSQLが混在する	依存関係解析＋設計との突合
4	**例外処理・入力値検証**	境界値・型・権限の検証、例外の握り潰しの不在、失敗時の状態整合	例外を包括捕捉してログのみ出力し処理継続する	人のレビュー＋異常系テスト
5	**重複・不要処理**	既存共通処理の再実装、到達不能コード、未使用変数・関数の残存	既存ユーティリティを使わず同等処理を新規実装	重複検出ツール＋人の確認
6	**外部ライブラリ**	追加の必要性、ライセンス、保守状況、既知脆弱性、バージョン固定	非推奨・更新停止のライブラリや旧APIを前提とする	依存監査（SCA）＋方針確認

禁止事項｜「AI生成コードだから」を理由にレビューを省略しない

・生成量が増えるほどレビュー負荷が上がるため、省略の圧力が働く。これに対しては**リスク区分（P6）による確認レベルの配分**で対処し、区分「高」の全行レビューは維持する。
・レビューAIの「指摘なし」は合格判定ではない。指摘の有無にかかわらず、承認は人が行う。
・差分が大きい生成物は、レビュー可能な単位へ分割して提出させる（1コミットあたりの変更行数の上限を規定する）。

G3（コード品質ゲート）通過条件

- ☐ 静的解析・Lintの重大指摘が0件
- ☐ 生成者以外のレビュー承認を取得
- ☐ 追加依存の妥当性と脆弱性を確認済み
- ☐ 設計・要件IDとの対応が記録済み
- ☐ 生成に用いたモデル・プロンプトを記録済み

注：変更行数上限等の閾値は組織規程で設定する。事実：生成AIは学習時点の実装様式を出力する／解釈：社内規約・既存共通処理との乖離が生じる／提案：6観点をレビュー基準として明文化する

セキュリティ品質の管理

AI駆動開発のセキュリティ管理は**「AIへ入力する情報の制御（入口）」と「AIが生成した実装の検証（出口）」の二方向**で構成する。
入口は運用ルールと技術的遮断で担保し、出口は下表の必須条件に該当する変更をセキュリティレビュー必須（GS）として扱う。

入口の管理｜AIへ何を入力させないか

① 機密情報・個人情報の入力禁止（仕組みで防ぐ）

- 入力禁止データを分類定義（個人情報、認証情報、非公開の契約・財務情報、未公開の設計情報）
- 利用可能なAIサービスを許可制とし、学習利用の有無を契約で確認
- マスキング・匿名化を経た検証用データのみ使用可とする
- ルール周知に加え、DLP・プロキシ制御等で技術的に遮断する

② 外部AIサービスへのデータ送信範囲の明確化

- 送信先サービス、送信されるデータ項目、保存期間、リージョンを台帳化
- ソースコード全体の送信可否を規程で明示（リポジトリ単位で可否を設定）
- セキュリティ担当の承認を経たサービス・用途のみ利用を許可

出口の管理｜生成された実装の何を検証するか

① 認証情報のハードコード防止 APIキー・パスワード・接続文字列の直書き、テストコードや設定ファイルへの残存を検出（シークレットスキャンをCIで必須化）

② 入力値起因の脆弱性対策 SQLインジェクション（プレースホルダの使用）、XSS（出力エスケープ）、パス・コマンドの外部入力連結の排除

③ アクセス権限の最小化 付与された権限が処理に必要な範囲か、管理者権限・広域スコープの既定利用がないかを確認

④ 依存ライブラリの脆弱性確認 追加・更新された依存を既知脆弱性データベースで照合し、バージョンを固定

⑤ ログ・エラー出力の情報漏えい防止 個人情報・トークン・内部構造がログや画面のエラー表示に出力されないことを確認

セキュリティレビュー（GS）が必須となる変更条件

変更の内容	必須となる確認行為	未実施の場合の扱い
認証・認可・権限判定ロジックの追加または変更	全行レビュー＋権限別テスト	G5に進めず差し戻し （例外リリースはP15の承認・記録ルールに従う）
個人情報・機密情報を取得・保存・出力する処理	データ項目の妥当性＋保存期間の確認	
外部公開API・外部システム連携の新規追加	送信データ範囲＋通信・認証方式の確認	
本番データを直接操作する処理・バッチ・移行スクリプト	実行手順・復旧手段の確認＋立会い実行	

注：入力禁止データの分類は自社の情報管理規程に整合させる。事実：外部AIサービス利用時はデータが社外へ送信され得る／解釈：周知のみでは入力を防げない／提案：技術的遮断と許可制を併用する

AI生成テストの品質管理

AIが生成したコードを同じAIだけで評価すると、**誤った仕様理解がテストによって「正解」として固定される。**
したがってテストは仕様（要件ID）から独立に導出し、正常系偏重を避けるため観点別の必須内訳をG4の判定条件として定義する。

1. テスト観点の内訳と生成時の指示方針

観点	AIへの生成指示に含める内容	人による確認の要点
正常系	受入条件の各項目を1件以上のケースに対応させる	要件IDとの対応の網羅
異常系	入力不正、必須欠落、外部連携の失敗・タイムアウト、重複実行	業務手順との整合 （画面表示と後続処理）
境界値	上限・下限とその±1、桁数・文字種、日付境界、0件・最大件数	仕様上の境界値の妥当性
権限別	役割ごとの許可・拒否、他者データへの参照、権限昇格の試行	拒否時の挙動 （セキュリティ担当と確認）
回帰	既存機能への影響範囲、過去不具合の再発防止ケースの追加	自動化と実行頻度 （CIでの必須実行）

2. 同一AIによる自己評価の危険性と対処

避けるべき構成｜生成と検証が同一前提

同一モデル・同一プロンプト前提でコードとテストを生成すると、仕様の誤解がテストの期待値にも反映され、**誤りが「全件成功」として通過する**。カバレッジは上昇するため、数値上は問題が見えない。

対処｜独立性の確保（3点）

① テストは**コードではなく仕様（要件ID・受入条件）から**導出させる
② テスト生成は開発AIと別のエージェント・別基準で実施（P13）
③ テストコード自体をレビュー対象とし、期待値の根拠を人が確認

3. G4（テスト品質ゲート）通過条件とテストコードのレビュー観点

G4通過条件

- ☐ 全要件IDにテストケースが対応（対応表を提出）
- ☐ 異常系・境界値・権限別のケースが観点ごとに存在する
- ☐ 回帰テストがCIで自動実行され、全件成功
- ☐ カバレッジが規定値以上、かつ未到達箇所の理由が記録済み
- ☐ テストコードのレビュー承認（生成者以外）を取得

テストコード自体のレビュー観点

- ☐ 期待値が仕様に基づくか（実装の出力をそのまま期待値にしていないか）
- ☐ 常に成功する記述、無意味な検証、検証の欠落がないか
- ☐ 実行順序・共有データへの依存がないか（単独実行で再現するか）
- ☐ 本番データ・実認証情報を用いていないか
- ☐ カバレッジのための形式的なケースになっていないか

注：カバレッジ規定値は対象システムの重要度に応じ設定する。事実：テストの期待値は生成時の仕様理解に依存する／解釈：同一前提の自己検証は誤りを追認する／提案：仕様起点の導出と別基準での生成を必須とする

品質を高める複数AIエージェント構成

開発・レビュー・テストを**異なる指示と評価基準を与えた別エージェントに分担させる**ことで、単一エージェントでは検出されない見落としを相互に補完できる。
ただしAI同士の合意は合格判定ではなく、承認は常に人が行う構成を前提とする。

① 開発AIエージェント

役割　承認済み設計に基づく実装・修正・リファクタリング

与える指示(例)
・設計書と要件IDの範囲内で実装する
・社内コーディング規約と既存共通処理を優先する
・設計に記載のない仕様は実装せず、確認事項として列挙する

評価基準　要件充足・規約適合・既存構成との整合

② レビューAIエージェント

役割　実装の欠陥候補・規約違反・脆弱性の指摘

与える指示(例)
・実装の意図を推測せず、設計・要件との差分のみを根拠に指摘する
・指摘には該当箇所・理由・想定影響を必ず付す
・「問題なし」と判断した観点も明示して報告する

評価基準　欠陥検出・脆弱性・保守性(開発側と別基準)

③ テストAIエージェント

役割　仕様起点のテストケース・テストコード生成

与える指示(例)
・**実装コードを参照せず**、要件・受入条件のみを入力とする
・異常系・境界値・権限別を観点別に必ず生成する
・期待値の根拠となる要件IDを各ケースに付す

評価基準　仕様網羅・観点網羅・再現性

基本構成｜相互レビューと人による承認の位置

① 開発AI 実装生成	② レビューAI 指摘・差分検証	③ テストAI 仕様起点で検証	指摘・不一致の集約 根拠付きで提示	④ 人による承認 レビュー担当／最終承認者
設計・要件IDの範囲内で生成し、確認事項を明示	開発側と別の基準・別プロンプトで検証(同一設定を禁止)	実装非参照。テスト失敗は実装・仕様いずれの問題か切り分ける	未解消の指摘・不一致は残存リスクとして記録	指摘の採否を判断し、G3・G4の通過を決定。判断根拠を記録する

禁止事項｜AI同士の合意を最終承認としない

レビューAIとテストAIが「問題なし」と判定しても、それは**人の確認範囲を絞り込む情報**であり、合格判定ではない。同一モデル・同一前提での多重確認は独立性を持たない。

独立性を高める設定:レビューとテストで**異なるモデル／異なる評価基準／異なる入力範囲**を用いる。少なくとも入力範囲(実装参照の可否)は必ず分離する。

注:指示文は例示であり、対象システムに応じて具体化する。事実:同一前提の検証は同じ誤解を共有する／解釈:独立性の源泉は基準と入力範囲の差／提案:実装参照の可否を分離し、承認は人が行う

Human-in-the-Loopの設計

人の介在は「余力があるときに確認する」運用では機能しない。**要件確定・設計決定・本番データ・セキュリティ・リリース前の5点を必須介在点として固定し、**
それ以外はリスクレベル別の承認フローに委ねることで、確認負荷を集中させつつ重要判断の抜けを防ぐ。

1. 人が必ず介在すべき5つのポイント

① 要件確定時	② 基本設計・方式決定時	③ 本番データを扱う変更時	④ セキュリティ関連変更時	⑤ 本番リリース前
【AI推測】の可否判断、受入条件の合意、【未決】の残存確認 **承認：業務・仕様責任者** 記録：合意日・対象要件ID	設計案の選定と選定理由、既存構成への影響、残存リスクの受容判断 **承認：プロジェクト責任者** 記録：比較表・選定理由	データ移行・一括更新・削除処理の内容、影響件数、復旧手段、実行の立会い **承認：最終承認者** 記録：実行計画・実行結果	認証・認可・権限、個人情報の取扱い、外部連携の追加（P11のGS条件） **承認：セキュリティ担当** 記録：確認結果・残存リスク	G1～G4・GSの通過状況、残存リスク、切り手順、監視体制 **承認：最終承認者** 記録：判定結果・判断根拠

2. リスクレベル別の承認フロー

区分	承認フロー	AIの自動実行	記録保存
高	生成 → 人の全行レビュー → セキュリティ確認 → 最終承認者の承認（2名以上）	不可（各段で停止し、人の承認を待つ）	全項目（P17）＋立会い記録
中	生成 → レビューAIの指摘対応 → 人の差分レビュー承認（生成者以外1名）	検証環境まで可。本番反映は人の承認後	全項目（P17）
低	生成 → 自動検査（Lint・テスト）通過 → 差分の要点確認（1名）	自動検査の範囲で可	モデル・プロンプト・差分

3. AIによる自動実行を停止する条件

以下に該当した時点で処理を停止し、人へエスカレーションする

① 設計・要件に記載のない仕様を実装する必要が生じた場合
② 本番環境・本番データへの書き込みまたは削除を伴う場合
③ 認証・認可・権限・暗号化に関わる箇所を変更する場合
④ 依存ライブラリの追加、またはメジャーバージョン更新が必要な場合
⑤ 同一箇所の修正が規定回数を超えても是正されない場合
⑥ テストが失敗し、原因が実装・仕様のいずれか切り分けられない場合
⑦ 入力禁止データを参照する必要が生じた場合（即時停止・記録）

注：承認者数・修正回数の上限は組織規程で設定する。事実：AIエージェントは連続実行により人の関与機会を減らす／解釈：介在点の事後判断は抜けを生む／提案：5点を固定の必須介在点とする

品質ゲートとリリース判定

品質ゲートは**「必須条件を満たさなければ次工程へ進めない」拒否権として運用してはじめて機能する。**
そのため各ゲートに必須条件・判定者・差し戻し先を定義し、例外リリースは承認者と記録を伴う手続きとして限定的に認める。

ID	ゲート	必須条件（1つでも未達なら不通過）	判定者	差し戻し先
G1	要件品質	【AI推測】が全件判定済み・【未決】が実装対象に含まれない・受入条件がテスト可能な記述	業務・仕様責任者	要件定義（業務部門との再確認）
G2	設計品質	全要件IDが設計要素に割当済み・2案以上の比較と選定理由の記録・外部連携仕様の実在確認	プロジェクト責任者 （＋セキュリティ担当）	設計（不整合が要件由来ならG1へ）
G3	コード品質	静的解析の重大指摘0件・生成者以外のレビュー承認・追加依存の妥当性確認・要件IDとの対応記録	AIレビュー担当者	実装（設計逸脱の場合はG2へ）
G4	テスト品質	全要件IDにケース対応・異常系／境界値／権限別が存在・回帰テスト全件成功・テストコードのレビュー承認	品質管理担当	実装またはテスト設計
GS	セキュリティ品質 （工程横断）	認証情報の直書き0件・既知脆弱性の未解消0件・権限最小化の確認・外部送信データ範囲の承認	セキュリティ担当	該当工程（設計由来ならG2へ）
G5	リリース品質	G1～G4・GSの全通過・残存リスクの明示と受容判断・切り戻し手順の確認・監視項目と通知先の設定完了	最終承認者	未達ゲートへ差し戻し（リリース保留）

必須条件が未達の場合の差し戻し

・未達項目・原因工程・是正内容・再判定日を記録し、**原因が発生した工程まで戻す**（実装のみの修正で塞がない）
・同一ゲートで規定回数以上不通過となった場合は、要件・設計の前提またはプロンプト設計を見直す
・差し戻し理由は指摘件数として集計し、P16の指標として傾向を管理する

例外リリース時の承認・記録ルール

・例外は「業務停止回避等の緊急時」に限り、**プロジェクト責任者と最終承認者の連名承認**を必須とする
・未達条件、残存リスク、暫定対処、恒久対応の期限と担当を記録し、監視強化の対象とする
・**GSの必須条件未達は例外の対象外**（セキュリティは例外リリースを認めない）

注：不通過回数の上限等の閾値は組織規程で定める。事実：ゲートは判定者と必須条件がなければ形式化する／解釈：拒否権を伴わない確認は通過儀礼となる／提案：G1～G5＋GSに判定者と差し戻し先を明記する

AI駆動開発の品質評価指標

生産性指標のみを追うと、品質低下が手戻り・障害として後から現れる。**生産性と品質の指標を同一期間で対にして評価し、**
片方のみが改善している状態を異常として検知する運用とする。

生産性｜AI生成コードの採用率

70% （目標値・仮定値）

採用行数 ÷ 生成行数。低下時は指示・設計の精度不足を示す

品質｜人による修正率

20%以下 （目標値・仮定値）

修正行数 ÷ 採用行数。上昇は仕様理解の不足を示す

品質｜本番障害件数

導入前以下 （必須条件）

重大度別に集計。増加時はAI適用範囲を縮小し原因を分析

生産性｜AI利用による削減工数

測定値 （検証・修正工数を控除）

生成による短縮分から、レビュー・手戻り増分を差し引いて算出

評価指標の一覧｜定義・測定元・判断の視点

分類	指標	定義・算出方法	測定元	判断の視点
生産性	AI生成コードの採用率	採用行数 ÷ 生成行数	生成ログ／コミット差分	低い＝指示・設計の精度不足
	AI利用による削減工数	従来実績比の短縮工数 −（レビュー＋手戻り増分）	工数管理／課題管理	控除後で評価（総所要時間で判断）
	手戻り工数	ゲート差し戻し・再作業に要した工数	ゲート判定記録／工数管理	発生工程の偏りを分析
品質	人による修正率	修正行数 ÷ 採用行数	コミット差分	上昇＝仕様理解・指示の不足
	レビュー指摘件数／不具合検出率	指摘件数（重大度別）／工程内検出数 ÷ 総検出数	レビュー記録／不具合票	工程内検出比率の向上を目標とする
	テストカバレッジ	網羅率（観点別の内訳と併記）	CI測定結果	数値単独では判断しない（P12参照）
	本番障害件数／脆弱性検出件数	重大度別の件数と未解消件数	障害記録／脆弱性診断結果	増加時はAI適用範囲を縮小し原因分析

注：採用率70%・修正率20%は目標設定の記述例（仮定値）であり、実測に基づき自社基準を設定する。事実：生産性と品質は別指標／解釈：片方のみの改善は品質の繰延べを示す／提案：対で同期間に評価する

ログ・証跡・変更履歴の管理

監査や障害調査で問われるのは「なぜこの実装になったか」であり、**入力・生成物・修正・承認を一連の記録として残せているか**が説明可能性を決める。
そのため記録は8項目を必須とし、コード・テスト・設計書と相互に辿れる形で保存する。

1. 記録必須項目（8項目）と保存方法

#	記録項目	記録内容	保存先の例	説明できるようになる問い
1	AIモデルとバージョン	サービス名、モデル名、版、利用形態（社内／外部）	コミットのメタ情報／生成ログ	何を使って生成したか
2	入力プロンプト	指示文、システム指示、評価基準、再生成時の差分	プロンプト管理リポジトリ	どう指示したか
3	参照した仕様書・ソースコード	資料名と版、参照範囲、対象リポジトリとリビジョン	生成ログ／構成管理	何に基づく判断か
4	AIが生成した内容	生成物の原本（修正前）、AIが列挙した確認事項・前提	生成物保管領域／初回コミット	AIは何を出力したか
5	人が修正・承認した内容	修正差分と理由、レビュー指摘とその採否、承認結果	レビュー記録／承認記録	人はどこを判断したか
6	実行日時と実行担当者	生成・修正・承認の各日時、実行者、エージェント起動者	操作ログ／構成管理履歴	誰がいつ実施したか
7	成果物との紐付け	要件ID・設計書・コミット・テストケースの相互参照	課題管理／構成管理（ID連携）	どの要件の実現物か
8	ゲート判定結果	G1〜G5・GSの判定、未達項目、残存リスクと受容判断	品質記録／リリース判定記録	なぜリリース可としたか

2. 監査時に説明できる状態｜双方向に辿れる連鎖



保存にあたる留意点：プロンプト・生成ログには機密情報が含まれ得るため、アクセス権限を限定し保存期間を規定する（記録が新たな情報漏えい経路にならないようにする）。
外部AIサービス利用時は、事業者側のログ保存期間・学習利用の有無を契約で確認し、台帳（P11）と併せて管理する。

注：保存期間・保存先は自社の文書管理規程に整合させる。事実：生成物の妥当性は入力と参照資料に依存する／解釈：記録欠落時は経緯を再現できない／提案：8項目を必須記録とし成果物IDで連結する

導入時によくある失敗と対策

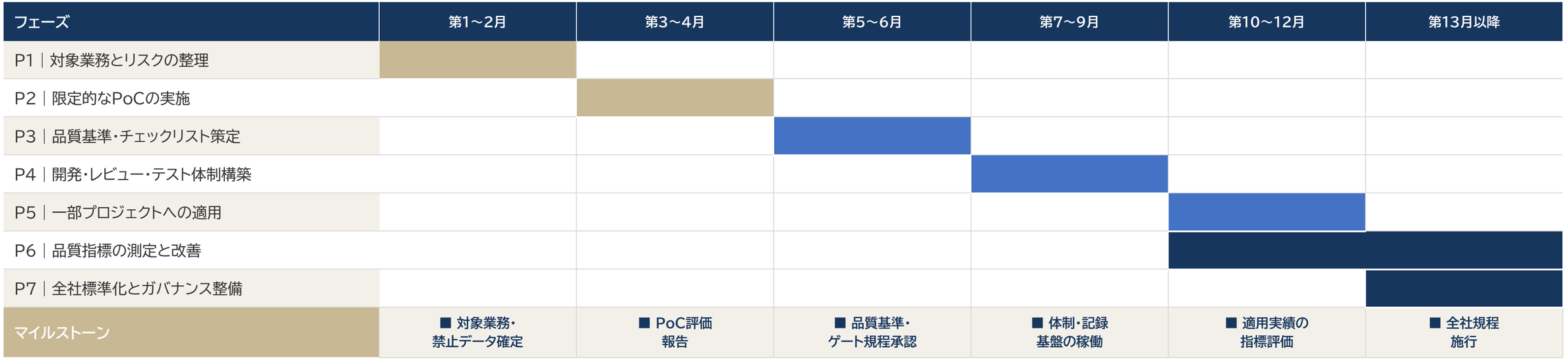
失敗の多くは技術要因ではなく、**判断の委譲・検証の省略・記録の欠落・責任の不明確さという運用設計の欠陥**に起因する。
したがって8類型それぞれに対し、規程・体制・仕組みのいずれかで是正する対策を、導入前に定めておく必要がある。

#	よくある失敗	発生する事象	対策	参照
1	AIに要件判断まで任せる	推測で補完された仕様が確定仕様として実装され、受入時に業務と齟齬が判明する	【確定】【AI推測】【未決】のラベル運用を要件一覧の必須列とし、G1で全件判定する	P8・P15
2	出力されたコードをそのまま本番へ反映する	脆弱性・非推奨実装・保守困難な構造が残存し、障害と改修費として顕在化する	リスク区分別の確認レベルを規程化し、区分「高」は全行レビューと2名承認を必須とする	P6・P10
3	レビューAIと開発AIの評価基準が同じ	同一の誤解を共有し「指摘なし」と判定され、欠陥が合格として通過する	レビューは別プロンプト・別評価基準で実施し、少なくとも入力範囲（実装参照の可否）を分離する	P13
4	AI生成テストだけで品質を判断する	カバレッジは上昇する一方、異常系・境界値・権限別が不足し本番で初発の不具合が出る	テストは仕様起点で導出させ、観点別の必須内訳とテストコードのレビューをG4条件とする	P12
5	品質指標を設定せず速度だけを評価する	短期の効率化が評価される一方、手戻り・障害増加が見えず、後期に効果が相殺される	生産性と品質の指標を対で同期間に測定し、片方のみの改善を異常として扱う	P16
6	プロンプトや生成履歴が残っていない	障害調査・監査で経緯を再現できず、同種の生成物の影響範囲も特定できない	記録必須8項目を定め、成果物IDで連結して保存する（保存期間・権限も規定）	P17
7	責任者と承認者が不明確	生成者が自らの成果を承認し、AIエージェントの判断が実質的な決定となる	RACIで各工程のAを1名に限定し、生成者と承認者の兼務を禁止する	P7・P14
8	小規模検証をせず全社導入する	基準もチェックリストもない状態で利用が拡大し、品質のばらつきが統制不能になる	PoCで基準を策定し、限定適用と測定を経てから標準化する段階的導入とする	P19

注：8類型は本資料の整理であり、優先順位は組織の状況により異なる。事実：失敗は運用設計の欠陥に起因する類型が多い／解釈：技術対策のみでは再発する／提案：規程・体制・仕組みのいずれかで是正する

段階的な導入ロードマップ

全社標準化を先行させると、基準のない状態で利用が拡大し統制不能になる。**低リスク業務から開始し、基準策定と実測を経て範囲を拡大する**7フェーズを推奨する。
各フェーズには次段階へ進む判定条件（Exit条件）を設定する。



凡例：■ PoC（検証） ■ 展開（基準策定・体制構築・適用） ■ 運用（測定・標準化） ■ マイルストーン

各フェーズの主要活動と次段階へ進む判定条件（Exit条件）

フェーズ	主要活動	Exit条件（未達なら次フェーズへ進まない）
P1・P2	対象業務の選定、AIへの入力禁止データ分類、限定範囲でのPoC実施	禁止データ分類の承認、PoCでの採用率・修正率の実測値取得
P3・P4	品質基準・ゲート条件・チェックリスト策定、RACI確定、エージェント構成と記録基盤の整備	G1〜G5・GSの必須条件と判定者の承認、記録8項目の自動取得の稼働
P5・P6	リスク区分「中」以下の案件へ適用、指標の月次測定と是正	本番障害件数が導入前以下、かつ手戻り工数が減少傾向にあること
P7	全社規程化、教育、外部AIサービス台帳の運用、監査対応の定着	規程の施行、全対象部門での記録取得率100%、監査での説明可能性の確認

注：期間（第1月〜第13月以降）は標準的な進行を示す想定値（仮定値）。事実：基準策定前の拡大は統制を欠く／解釈：範囲拡大の可否は実測で判断すべき／提案：Exit条件を満たすまで次段階へ進めない

まとめ・品質チェックリスト

AI駆動開発の品質は**最終確認・追跡可能性・役割分離・セキュリティ・試験網羅・記録・判定基準・稼働後監視の8点が満たされているか**で判断できる。
いずれかが「いいえ」の場合、その領域が最初に着手すべき是正対象となる。

#	確認項目	「はい」と言える状態の要件	確認者	参照
1	AIの出力を人が最終確認しているか	リスク区分別の確認レベルが規程化され、承認記録が残っている	レビュー担当	P6・P10
2	要件と生成物を追跡できるか	要件IDが設計・コミット・テストケースに付与され双方向に辿れる	PM／PMO	P8・P17
3	開発とレビューの役割が分離されているか	生成者は自らの成果を承認せず、レビューは別基準で実施されている	プロジェクト責任者	P7・P13
4	セキュリティ確認を実施しているか	入力禁止データが技術的に遮断され、GS必須条件が判定されている	セキュリティ担当	P11・P15
5	正常系・異常系・境界値をテストしているか	観点別の内訳が確認され、テストが仕様起点で導出されている	品質管理担当	P12
6	モデル・プロンプト・修正履歴を記録しているか	記録必須8項目が成果物IDで連結され、保存期間と権限が規定済み	PMO／品質管理担当	P17
7	リリース判定基準が明確か	G1～G5・GSの必須条件・判定者・差し戻し先が文書化されている	最終承認者	P15
8	本番稼働後の監視・改善体制があるか	障害・脆弱性・手戻りの指標が月次で測定され、是正が回っている	プロジェクト責任者	P16・P19

結論
Conclusion

AI駆動開発の成否を分けるのは生成の速さではなく、**生成物を検証し承認する仕組みが規程・体制・記録として整っているか**である。
まず着手すべきは、①ゲートの必須条件と判定者の明文化、②生成者と承認者の分離、③記録8項目の取得基盤の3点である。

お問い合わせ・コンサルティングのご案内
品質基準の策定、ゲート設計、体制構築、指標運用の支援を行っています。

Start Challenge Consulting co., ltd.
Mail：info@startchg.com ／ 担当：平出

初回相談（60分・無償）
詳細は次ページ（P21）をご参照ください

注：初回相談の条件は2026年8月時点の内容。事実：本資料は8領域の管理要件を定義した／解釈：未整備領域が品質リスクの発生源となる／提案：ゲート明文化・役割分離・記録基盤の3点から着手する

お問い合わせ・ご支援メニュー

品質管理の仕組みは、規程の文書化だけでは定着しない。**基準策定・体制構築・運用定着までを一連で支援する**ことを方針としています。
まずはP20のチェックリスト8項目の診断から、現状のどの領域に着手すべきかを整理します。

お問い合わせ先

Start Challenge Consulting co., ltd.
挑む力が、世界を照らす

Start Challenge Consulting co., ltd.
誰も行かない、頂へ挑戦せよ

Mail	info@startchg.com
担当者	平出

ご支援メニュー

支援領域	主な内容	本資料の該当箇所
品質基準・ゲート設計	G1～G5・GSの必須条件と判定者の定義、差し戻し・例外リリース規程の整備	P4・P15
体制構築・役割分離	RACIの策定、生成者と承認者の分離、複数AIエージェント構成とHuman-in-the-Loopの設計	P7・P13・P14
レビュー・テスト運用	工程別チェックリストの作成、セキュリティレビュー条件の設定、観点別テストの内部化	P8～P12
指標運用・証跡整備	評価指標の設計と測定基盤の整備、記録必須8項目の取得と監査対応の定着	P16・P17・P19

ご相談の進め方

STEP 1　初回相談（60分・無償）	STEP 2　現状診断とご提案	STEP 3　支援の実施
現在のAI活用状況と課題を伺い、P20チェックリスト8項目で未整備領域を確認します。	優先着手領域、想定期間、体制案を整理し、支援範囲と進め方をご提案します。	基準策定・体制構築・運用定着まで伴走し、指標測定による改善サイクルの内部化までを支援します。

お問い合わせは **info@startchg.com**（担当：平出）まで。本資料に関するご質問・詳細版のご依頼も承ります。

注：初回相談の時間・費用条件は2026年8月時点の内容。事実：本資料は8領域の管理要件を定義した／解釈：未整備領域が品質リスクの発生源となる／提案：診断により着手順を確定する