

## 重大な矛盾/欠落 (P0)

- ADR承認フローの未定義が運用停止リスクを生む: Part00/Part14でADRを変更の必須先行とするが、承認者・方法・ツールが未定義。チーム規模拡大時やリモート運用で承認遅延/漏れが発生し、SSOT更新が停滞。2025-2026年のADRベストプラクティスでは、GitHub PRレビュー（最低2承認）や状態遷移（Proposed → Accepted）を標準推奨。
- Verify自動化の欠落が手動依存を招く: Part00の機械判定原則とPart10のVerify Gateが手動中心。2026年時点のGitHub Actionsベストプラクティスでは、PRマージ時に自動検証（linting/link check）を必須とし、手動忘れによる不整合流入を防ぐ。
- sources/不变性ルールの執行メカニズム不足: Part00で改変/削除禁止とするが、Git hooksやbranch protectionが未定義。重複ファイル扱い（ADR-0003）が曖昧で、誤操作時の復元が困難。Gitベストプラクティスでは、protected branchesとpre-commit hooksでimmutabilityを強制。

## 改善推奨 (P1/P2)

- P1: GitHub ActionsによるVerify/CI統合 → PR時にFast/Full Verifyを自動実行し、FAILでmerge block。markdownlintやカスタムスクリプトでリンク/用語揺れをチェック。セキュリティ強化にも寄与（第三者Actionのピニング推奨）。
- P1: SBOMツールの明確化と自動生成 → CycloneDXを優先（アプリケーション/供給チェーン向け）。syftやCycloneDX CLIを標準ツールに。NIST SSDF (SP 800-218) とNTIAガイドラインで両形式対応を推奨するが、CycloneDXのツールエコシステムが優位。
- P2: pre-commit hooksの導入 → ADR強制やsources/保護をクライアント側で実行。pre-commit frameworkでmarkdownlint/カスタムチェックを適用。用語揺れや禁止コマンドを自動検知。
- P2: ADRツールの拡張 → adr-toolsや専用テンプレートで状態管理（status: proposed/accepted）。チーム承認をPRレビューに紐付け。

## 具体的修正案 (Patch案)

・file: docs/Part00.md

**change:** 11. 未決事項 の U-0001 を解決セクションに移動し、以下を追加:

「ADR承認フロー: GitHub PRで最低2名のレビュー承認必須。status: proposed → accepted/deprecated。ツール: adr-tools CLIで状態管理。」

**reason:** ADRライフサイクルとPR統合を明確化し、承認漏れを防止。2025-2026ベストプラクティスに準拠。

・file: docs/Part10.md

**change:** 6. 手順 に新セクション「自動実行」を追加:

「GitHub Actionsワークフロー (.github/workflows/verify.yml) でPR時にFast/Full Verify自動実行。FAIL時はmerge block。例: actions/checkout + pwsh checks/verify\_repo.ps1。」

**reason:** 手動依存を解消し、再現性/事故ゼロを強化。GitHub Actions公式ガイドライン準拠。

・file: docs/Part14.md

**change:** 5. ルール の R-0003 (sources/関連) に追記:

「sources/保護: branch protection rulesでsources/への変更を制限 + pre-commit hooksで改変検知/ブロック。」

**reason:** 不変性ルールの執行力を高め、誤削除リスクを低減。Git immutabilityベストプラクティス準拠。

・file: docs/Part01.md

**change:** 5. ルール の R-0102 に追記:

「SBOM生成: CycloneDX優先。ツール: syft (コマンド: syft dir . -o cyclonedx-json > sbom.json)。SPDXもオプション出力。」

**reason:** ツール未定義を解消。2026年供給チェーンセキュリティ基準 (NIST/OWASP) に適合。

・file: docs/Part02.md

**change:** 8. 機械判定 に新V項追加:

「V-0202: 用語揺れ自動検知 - pre-commit + markdownlint custom rulesでglossary準拠チェック。」

**reason:** 用語統一の自動化で運用負荷軽減。

## “運用で事故る” 代表シナリオ3つ + 予防策

- ・シナリオ1: ADRなしの緊急docs/変更 → SSOT汚染/ロールバック困難

予防策: pre-commit hooksでdocs/変更時にdecisions/追加を強制 (pre-commit framework + カスタムスクリプト)。PR必須ブランチポリシー適用。

- ・シナリオ2: Verify手動忘れ → 不整合（リンク切れ/用語揺れ）マージ

予防策: GitHub ActionsでPR時に自動Verify実行 + required status check設定。FAILでmerge block。

- ・シナリオ3: sources/誤変更/削除 → 証拠喪失/監査不能

予防策: main branch protectionでsources/変更を制限 + git hooksで変更検知/拒否。バッカアップブランチ自動作成。

## 未決事項リスト + 確認手順

- ・U-0001/U-1401: ADR承認フロー詳細（承認者数/ツール）

確認手順: GitHubリポジトリ設定 (settings/branches) でbranch protection rules確認。adr-toolsをローカルインストールし、status管理テスト。チームレビューで最低承認数を決定。

- ・U-0004/U-1402: Verify自動化タイミングとツール

確認手順: .github/workflows/verify.yml サンプル作成 → PRイベントでテスト実行。

GitHub Actionsログ確認 (actions tab)。手動 vs CIの負荷比較をevidence/metricsに記録。

- ・U-0102: SBOM形式/ツール優先

確認手順: syftインストール → CycloneDX/SPDX両生成テスト。出力比較 (ツール: cyclonedx-cli validate)。NIST SSDF準拠度を評価。

- ・U-1403: CHANGELOG長大化対策

確認手順: CHANGELOG.md行数計測スクリプト実行 (wc -l)。1000行超で年別分割テスト → 検索性/編集性をevidenceに記録。ADRで方針決定。