

Contents

NFR_EXITON_TRADING_v0.1.md	1
EXITON 自動投資システム – 非機能要件 v0.1	1
1. パフォーマンス	1
NFR-TR-001 バックテスト応答時間	1
NFR-TR-002 シグナル再計算応答時間	1
2. 信頼性・壊れ方	2
NFR-TR-010 Silent Failure 禁止	2
NFR-TR-011 一貫性のある結果	2
NFR-TR-012 部分障害時の扱い	2
3. ログ・再現性・監査	2
NFR-TR-020 バックテストrun_id	2
NFR-TR-021 ログの標準形式	2
NFR-TR-022 監査ログ（将来）	3
4. セキュリティ・データ保護	3
NFR-TR-030 現フェーズの前提	3
NFR-TR-031 APIキーの扱い	3
NFR-TR-032 ネットワーク公開	3
5. 可用性・運用	3
NFR-TR-040 環境分離の方針	3
NFR-TR-041 コンフィグ管理	3
6. テスト性	3
NFR-TR-050 自動テスト範囲	3
NFR-TR-051 テストデータセット	4
7. UX・説明責任（非機能的だが重要な観点）	4
NFR-TR-060 リスクの見える化	4

NFR_EXITON_TRADING_v0.1.md

EXITON 自動投資システム – 非機能要件 v0.1

本ドキュメントは、EXITON 自動投資システム（ai-signal-chart）の **非機能要件（NFR: Non-Functional Requirements）** を定義する。

機能仕様「何ができるか」ではなく、**どのような品質レベルで動く必要があるか**を対象とする。

ID 形式：NFR-TR-XXX。

1. パフォーマンス

NFR-TR-001 バックテスト応答時間

- 対象：日足ベースの単一銘柄バックテスト
 - 期間：最大10年（約2500営業日想定）
- 要件：
 - 通常負荷時、**10秒以内**に結果を返すことを目標とする。
- 将来的には：
 - コード・アルゴリズム最適化により 3~5秒を目指す。

NFR-TR-002 シグナル再計算応答時間

- 「最新データを取得して、シグナルだけ再計算する」操作は、

- 単一銘柄あたり **1秒以内** を目標とする。
 - 将来的に複数銘柄同時計算を行う場合、
 - 内部でバッチ処理や非同期処理を検討する。
-

2. 信頼性・壊れ方

NFR-TR-010 Silent Failure 禁止

- 予期せぬ例外・データ取得エラー・外部APIエラー等が発生した場合：
 - システムは **誤ったシグナルやバックテスト結果を静かに返してはならない。**
 - 原則として：
 - * エラーメッセージを表示
 - * 当該実行を無効とする
- 「よく分からないが一応数字は出た」という状態を避ける。

NFR-TR-011 一貫性のある結果

- 同じ戦略ID・バージョン、同じパラメータ、同じデータセットについて：
 - 実行のたびに **同一のバックテスト結果** が得られること。
- ランダム要素を含む戦略では、
 - シード値の固定もしくは結果の揺らぎを仕様として明記する。

NFR-TR-012 部分障害時の扱い

- 外部データソースに一時的な障害が発生した場合：
 - 影響を受けた銘柄・期間をユーザーに明示し、
 - 可能な範囲で結果を返すか、処理全体を中止するかを判断する。
 - 「一部のデータが欠損したまま平然と進む」形は避ける。
-

3. ログ・再現性・監査

NFR-TR-020 バックテストrun_id

- 各バックテスト実行には、一意な run_id を付与する。
- run_id に紐づけて、少なくとも以下を保存する：
 - 戦略ID・バージョン
 - パラメータセット
 - 銘柄・期間
 - データソース情報（可能ならハッシュやバージョン）

NFR-TR-021 ログの標準形式

- ログは少なくとも以下の項目を持つ：
 - timestamp (UTC)
 - level (INFO/WARN/ERROR)
 - run_id (任意)
 - symbol (任意)
 - strategy_id (任意)
 - message
- 構造化ログ (JSON形式) を優先する。

NFR-TR-022 監査ログ（将来）

- 将来的にユーザー アカウント 機能が実装された場合：
 - ユーザーのバックテスト 実行履歴・設定変更は監査ログとして保存する。
 - 「いつ、誰が、何を実行したか」が追跡可能であること。
-

4. セキュリティ・データ保護

NFR-TR-030 現フェーズの前提

- 現段階ではローカル環境での個人利用を前提とし、
 - メールアドレス・パスワード等の個人情報は扱わない。
- 将来のSaaS化時には、SAAS_STANDARD_v0.1.md および SECURITY_BASELINE_SAAS.md（将来作成）に従う。

NFR-TR-031 APIキーの扱い

- 市場データ取得等のAPIキーは：
 - .env 等の環境変数で管理し、
 - Git リポジトリに決して含めない。
- ログに APIキーを出力しない。

NFR-TR-032 ネットワーク公開

- 開発・検証段階では、ローカルまたは信頼できるLAN内のみでアクセスする。
 - 外部公開する場合は：
 - HTTPS (TLS) 化
 - 簡易な認証などを必須とし、SaaS基準に従う。
-

5. 可用性・運用

NFR-TR-040 環境分離の方針

- 初期フェーズ：dev 環境のみ（ローカル実行）。
- SaaS化フェーズでは：
 - dev / staging / prod の3つに論理的に分離することを前提とする。

NFR-TR-041 コンフィグ管理

- 環境依存の設定（APIエンドポイント・キー・DB URL）は、
 - すべて環境変数または設定ファイルから読み込む。
 - ソースコードに直書きしない。
-

6. テスト性

NFR-TR-050 自動テスト範囲

- 以下の領域には必ず単体テストを用意する：
 - ポジションサイジング
 - 手数料・スリッページ計算
 - 代表的な戦略（MAクロス、RSI等）
 - ルックアヘッドが発生していないことの検証
- バグ修正時には、該当バグを再現するテストを追加してから修正する。

NFR-TR-051 テストデータセット

- 外部APIに依存しない **固定テストデータセット** を用意し、
 - 単体テスト・回帰テストはこれを用いて実行する。
 - 実データを用いた E2E テストは別途行う。
-

7. UX・説明責任（非機能的だが重要な観点）

NFR-TR-060 リスクの見える化

- ドローダウン・最大損失・勝率など、
 - ユーザーが「戦略の危険度」を感覚的に理解できる指標を表示する。
 - 特にキルスイッチや最大リスクに関して：
 - UI 上でパラメータと効果を説明する。
-

以上。