

GOLD SMC EA — 総合仕様書 & 引き継ぎ書

Ver.1.1 (SMC統合構造・精密確定版)

作成日：2025年11月

対象：MetaTrader5 / XAUUSD (GOLD)

開発形態：単一EA構成（外部ファイル依存なし）

本ドキュメントは、GOLD SMC EA

のロジック定義、実装仕様、運用ルール、検証基準を包括した正式版技術仕様兼引き継ぎ書である。

0. 開発目的と設計思想

【目的】

GOLD市場特有の高ボラティリティ・急変動に対応した「構造認識型」自動売買EAを構築する。単なるインジケータ売買ではなく、市場構造（BoS / OB / 再テスト）とキッカーパターン認識に基づく供給・需要ゾーン理論の自動化を目指す。

【設計思想】

- シンプル・堅牢・高再現性
- 複数ファイル構成を避け、EA単体（.mq5）で完結
- バックテストと実運用の再現性を最重要視
- 将来のAI学習適応を見据え、データ構造・ログ構造を明確化

1. システム概要

【基本仕様】

- 通貨ペア：XAUUSD (GOLD)
- 時間足構造：D1 (方向確認) / H4 (構造判定) / H1 (執行)
- 主要構造要素：Break of Structure (BoS) / Order Block (OB) / 再テスト / キッカーパターン
- トレード方向：H4のトレンド方向と一致する方向のみ新規エントリーを許可
- トレード時間帯：ロンドン・NY時間限定 (GMT基準で自動補正)
- 実装方式：OnBarベース判定 (ティック依存ロジックは使用しない)
- バックテスト方式：Every tick based on real ticks
- ファイル構成：EA単体 (MQL5/Experts/kutaGOLD/GOLD_SMC_EA.mq5)

2. 構造ロジック（理論仕様）

2.1 H4構造判定（BoS）

【スイング検出】

- 前後20バーで高値・安値を比較し、相対的なスイング高値 / スイング安値を抽出する。
- 同値の扱い：同値はブレイクなし（静止扱い）とし、BoS判定の対象外とする。

【トレンド定義】

- 2回連続の高値・安値切り上げで上昇トレンド。
- 2回連続の高値・安値切り下げで下降トレンド。

【BoS確定条件】

- 終値がスイング値を $ATR(14) \times 0.1$ 以上明確に突破。
- その状態が 2本連続で上抜け / 下抜けとして維持される。

【状態管理】

- 候補BoS 確定BoS 無効BoS (EXPIRED)
- BoS構造体に state (ACTIVE/EXPIRED/TESTED/BROKEN) を持たせて管理する。

【表示（将来の可視化拡張用）】

- 候補：点線
- 確定：実線 + 矢印
- 無効：グレーアウト

2.2 Order Block (OB)

【OB生成条件】

- BoS確定直前3~5本の逆行足を走査する。
- 上昇BoSの場合：最後の下降足
- 下降BoSの場合：最後の上昇足

【OB範囲定義】

- ローソク実体範囲 (Body) のみをOBコア領域とする。
 - ヒゲは補助的表示とし、OBの価格帯としては採用しない。
- 大口約定価格帯 (支配的な実体領域) にフォーカスし、OBが過度に広がることを防ぐ。

【有効期限・テスト管理】

- 新鮮OB : 50本
- 古いOB : 100本
- テスト済OB : 3回テストで削除対象

【テスト判定】

- OB内に価格が侵入し、3本以内に反発した場合 : テスト成功。
- 5本以上停滞した場合 : 弱体化フラグを付与。

【複数OBの重複処理】

- 最新BoSに紐づくOBを優先し、古いBoS由来のOBは EXPIRED 化する。
市場の関心が最新の構造にシフトしている前提で、古いOBを整理しシステム負荷を軽減する。

【表示ルール（将来拡張）】

- 新鮮OB : 濃色
- 古いOB : 点線
- テスト済 : テスト回数に応じて「×1」、「×2」等を表示

2.3 キッカーパターン (Kicker)

【定義】

- 弱気ローソクの直後に、ギャップを伴う強気ローソクが出現し、かつ強い実体で前足を包み込むパターン（またはその逆）。

【検出条件】

- 2本間の終値と始値のギャップが、平均ATRの 0.5 倍を超える。
- 2本目の実体が1本目の実体を完全に包み、かつ下ヒゲ（または上ヒゲ）がほぼ存在しない。

【階層構造とスコアリング】

1. BoS確定時 : 構造信頼性向上 (+50pt)
2. OB生成時 : OB品質評価 (+30pt)
3. OB再侵入時 (再テスト) : トリガー条件 (+20pt)

【スコア加算ルール】

- 各層 (BoS, OB, 再テスト) は独立した根拠として扱い、それぞれの条件を満たした場合にのみ加算する。
- 内包関係にある要素（例：同一OBに対する再テストの繰り返し）では二重カウントを防止するロジックを実装する。

【スコアとリスクの対応】

- スコア 80以上 : リスク 2% (フルポジション)
- スコア 60~79 : リスク 1.5% (標準)
- スコア 50~59 : リスク 1.0% (軽量)
- スコア 50未満 : 新規エントリー見送り

2.4 再テスト条件

【価格帯】

- BoS_Level \pm (ATR \times 0.25) 範囲での再接触を再テスト候補とする。

【判定精度】

- 高精度 : \pm ATR \times 0.1 即エントリー許可
- 標準 : \pm ATR \times 0.25 キッカーパターン併用を推奨
- 中精度 : \pm ATR \times 0.5 補助条件 (時間帯・ボラ状況) 必須
- 低精度 : \pm ATR \times 0.75 原則無効化推奨

【回数制限】

- 1回目 : 信頼度100%
- 2回目 : 信頼度70% (10本以上の間隔がある場合に限定)
- 3回目 : 信頼度40% (原則エントリー見送り)

2.5 エントリー条件

エントリーは以下の条件を満たした場合のみ許可される。

【必須条件】

- H4トレンド方向とエントリー方向が一致していること。
- H1において、BoSレベルまたはOBコア領域の再テスト後、反発サインが確認されていること。

【補助条件 (少なくとも1つ以上)】

- キッカーパターンの出現。
- ATR拡大中 (ボラティリティ上昇局面)。
- ロンドン / NYのコア時間帯に一致。

これらを総合スコアとして評価し、所定のスコア閾値 (50以上) を満たした場合のみ、新規ポジションを構築する。

2.6 エグジット条件

【Exit条件の優先順位】

1. BoS逆転 (構造的条件・最優先)
2. OBブレイク (需給バランス崩壊)
3. ATRベースのリスク管理・時間制限

【構造的条件】

- 逆方向のBoSが確定した場合 : 全ポジションを即時全決済。
- 既存ポジション方向の主要OBが明確にブレイクされた場合 : 50%決済。

- 上位時間軸 (D1/H4) で逆BoSが発生した場合 : 75%決済 + トレールを強化。

【リスク条件】

- 初期ストップロス : $ATR \times 1.5$
- トレーリングストップ : $ATR \times 1.0$ を基準としたダイナミックトレール。
- アジア時間跨ぎ / 週末2時間前 : ポジション縮小または全決済を強制。

【利益確定条件】

- TP1 = $ATR \times 1.5$ (ポジションの30%を利確)
- TP2 = $ATR \times 3.0$ (さらに30%を利確)
- TP3 = $ATR \times 5.0$ (さらに30%を利確)
- 残り10%はトレールに委ね、トレンドフォロー型のエクステンションを狙う。

3. 実装上の主要対策 (7大問題対応)

以下の7つの問題に対して、あらかじめ対策を講じる。

1. 時間軸同期ズレ

階層バー確定 + 同期キュー方式 / 遅延評価で、H4・H1・M15などのバー確定順序を統一。

2. データ肥大化

循環バッファ (リングバッファ) + 状態ベース削除で、古いBoS/OB/ログを自動整理。

3. ATR・出来高異常

多層フォールバック + 異常値除外 (スパイク検出) で、異常ボラティリティ時の誤判定を防止。

4. 重複BoS/OB

UUID (識別子) / 親子リンク / フラグ管理により、重複構造の発生と混線を防ぐ。

5. セッションズレ

GMT/DST自動補正 + 境界15分の新規エントリー停止で、ロンドン・NYセッション切替時のブレを抑制。

6. 週末ギャップ誤判定

$ATR \times 3$ 以上のギャップを検出し、初動バーは待機モードとして構造判定対象外とする。

7. バックテスト差異

Every tick based on real ticks + OnBar統一処理 +
浮動小数点誤差補正で、バックテストとデモの乖離を最小化する。

4. コード骨格 (テンプレート構造)

```
//+-----+
//| GOLD_SMC_EA.mq5 (Core Template)
//+-----+
#property strict
#include <Trade/Trade.mqh>
CTrade trade;

//===== Input Settings =====/
input double Lots = 0.10;
input int ATR_Period = 14;
```

```

input int Swing_Len_H4 = 20;
input double RiskPerTrade = 1.5;

//===== ENUMS =====/
enum STATE { ACTIVE, EXPIRED, TESTED, BROKEN };

//===== STRUCTS =====/
struct BoS {
    string id;
    datetime time;
    double level;
    bool bullish;
    STATE state;
    double atr;
    int score;
};

struct OB {
    string id;
    datetime time;
    double high, low;
    bool bullish;
    STATE state;
    int testCount;
    string parentBoS;
};

//===== GLOBAL ARRAYS =====/
BoS bos_list[50];
OB ob_list[100];

//===== FUNCTION PROTOTYPES =====/
void CheckBoS();
void CheckOB();
void CheckKicker();
void EvaluateEntry();
void ExitLogic();
double GetATR(string symbol, ENUM_TIMEFRAMES tf, int period);

//===== MAIN ENTRY =====/
int OnInit() { Print("GOLD SMC EA initialized."); return(INIT_SUCCEEDED); }

void OnTick() {
    UpdateTrailingStop();
    MonitorEmergencyExit();
}

int OnCalculate(...){
    static datetime last_bar=0;
    datetime bar_time=iTime(_Symbol,_Period,0);
    if(bar_time!=last_bar){
        last_bar=bar_time;
        CheckBoS();
        CheckOB();
        CheckKicker();
        EvaluateEntry();
    }
    return(0);
}

```

5. 構造フロー : BoS OB Kicker Entry

【基本フロー】

[BoS検出]

[確定BoS]

OB生成
(同時にキッカー確認)

[プレミアムOB] スコア+30

[再テスト監視]

[キッカー確認層3]

[エントリー実行]

[Exit監視 (BoS逆転 / ATR / 時間制限)]

このフローにより、構造認識 (BoS) 需給ゾーン (OB) モメンタム (Kicker)
エントリー / エグジットが一貫した論理で接続される。

6. 運用管理ルール

【稼働時間帯】

- 16:00 ~ 02:00 JST (ロンドン ~ NYの高流動時間帯をカバー)。
- 実際の実装では
GMT基準でロンドン / NYセッションの開始・終了を計算し、DST (サマータイム) を自動補正する。

【スプレッド管理】

- スプレッド >
100ポイント (ブローカー仕様に応じて適宜換算) の場合 : 新規エントリーを自動スキップ。

【重要イベント回避】

- 雇用統計・FOMCなどの主要指標の発表前後30分間は新規エントリーを停止する (実装時は経済指標レンダー連携または手動設定)。

【週末リスク管理】

- 金曜NY時間 12:00 (日本時間 2:00) 時点で全ポジションを決済し、週末ギャップリスクを回避する。

7. バックテスト・品質検証

【目標値】

- バックテストとデモの乖離 : ±5%以内
- 勝率 (BoS一致時) : 65%以上
- RR比平均 : 1 : 2.5 以上
- 最大ドローダウン (DD) : 15%以下

- トレード頻度：月20～30回（標準）
- ログ一致率：95%以上

【基準値の定義】

- 特定期間（推奨：直近12ヶ月）のデモ環境における「月次損益の中央値」を基準値とする。
- 乖離率は以下の式で定義する：

$$\text{乖離率} = |\text{バックテスト損益} - \text{デモ中央値}| \div \text{デモ中央値} \times 100 (\%)$$

- 平均値ではなく中央値（メディアン）を採用することで、外れ値に対する耐性を高め、評価指標の堅牢性を確保する。

8. 補足仕様 (Ver.1.1 精密確定版 追記)

以下の7項目は、本EAの再現性・一貫性を高めるために Ver.1.1 で明確化された補足仕様である。

1. BoSのスイング認識ロジック

- 同値はブレイクなし（静止扱い）。
- 厳格なブレイク定義により、レンジ相場でのノイズ発生を防止。

2. Order Block (OB) の範囲定義

- ローソク実体範囲 (Body) のみをOBとして採用。
- ヒゲは補助的表示とし、価格帯には含めない。

3. 複数OBの重複処理

- 最新BoSに紐づくOBを優先。
- 古いOBは EXPIRED 化して無効化。

4. Kickerスコアの多層加算

- BoS、OB、再テストなど各層を独立根拠と見なし、条件を満たした層のみスコア加算。
- 同一OB内での再テストなど、内包関係にある要素は二重カウントを防止。

5. 時間帯とセッション補正

- GMT基準でロンドン・NYの時差を自動補正。
- サマータイム有無にかかわらず、常にコアタイム（高流動帯）を正確に捉える。

6. Exit条件の優先順位

- BoS逆転 > OBブレイク > ATR / 時間制限。
- 構造崩壊 (BoS逆転) を最優先に検知し、最大損失回避を重視。

7. 検証ステップ：基準値の定義

- 基準値にはデモ環境12ヶ月間の「月次損益中央値」を採用。
- ±5%乖離の評価を現実的な安定指標として扱う。

9. 開発ロードマップ (再掲)

【フェーズ別ロードマップ】

1. 基盤構築 (1 ~ 2週)

- OnBar統一処理の実装。
- GMT / DST補正ロジックの実装。
- テスト環境 (デモ口座・バックテスト設定) の整備。

2. データ管理 (1週)

- 循環バッファ構造の実装。
- BoS / OB / ログに対するUUID実装。

3. 信頼性強化 (1週)

- ATR異常値対策。
- 週末ギャップ除外ロジックの実装。

4. 統合テスト（1～2週）

- 結合テスト・パラメータ最適化。
- バックテストとデモの再現性検証（±5%以内）。

10. 引き継ぎメモ

【ソース情報】

- ファイル名 : GOLD_SMC_EA.mq5
- 配置パス : MQL5/Experts/kutaGOLD/

【依存ファイル】

- のみ (外部ライブラリ依存なし)。

【バックテスト設定】

- モデル : Every tick (real ticks)
- データ期間 : 直近1年
- 使用時間足 : H1 または H4

【主要関数呼び出し順序】

- OnCalculate CheckBoS CheckOB CheckKicker EvaluateEntry ExitLogic
- OnTick では、UpdateTrailingStop / MonitorEmergencyExit を実行。

【将来拡張ポイント】

- FVG (Fair Value Gap) 検出口ジックの追加 (Optional)。
- Sweep判定 (Liquidity Grab) の導入。
- AI適応スコアリング (Phase 5予定 : PhaseScoreなどとの連携)。

11. 結論

本ドキュメントは、GOLD SMC EA に関する

- ロジック定義 (理論 + 実装)
- 実行構造とフロー
- エラー・リスク対策
- テスト・品質評価基準
- 引き継ぎ運用手順

をすべて包括した、正式版技術仕様兼引き継ぎ書である。

本仕様書を前提とすれば、他者がEAを再構築・保守・最適化する場合でも、ロジック・挙動を100%再現することが可能である。

本Ver.1.1は、今後のバージョン展開 (Ver.2.x以降) およびAI連携拡張のための安定した基盤仕様として位置づける。