

# Die Bot Safety Checklist.

14 Fragen, die entscheiden, ob Ihre Handelsautomatisierung einen echten Markt überlebt.

---

## Pavlo Filianov

Gründer, FAMBF — Framework for Automated, Mandate-Backed Finance

~30 MIN LESEZEIT

14 FRAGEN · 0–14 PUNKTE

---

## 00 · WARUM ES DAS GIBT

# Einleitung.

Ich habe diese Checkliste angefangen zu schreiben, nachdem der dritte Trader, den ich persönlich kannte, den Großteil eines Kontos durch einen Bug in seinem eigenen Bot verloren hatte. Keiner von ihnen verlor das Geld an den Markt. Sie verloren es an etwas in der Automatisierung, das hätte aufgefangen werden müssen, bevor das System je an echtes Kapital angeschlossen wurde.

Das Muster ist immer dasselbe. Die Strategie passt, der Code funktioniert unter normalen Bedingungen, und dann passiert etwas Ungewöhnliches. Ein Reconnect im falschen Moment. Ein doppelt eintreffendes Signal. Ein Webhook, der zwanzig Minuten zu spät kommt. Der Bot tut etwas, das der Trader nie beabsichtigt hat, und wenn schließlich jemand in die Logs schaut, ist der Schaden längst entstanden.

Das ist eine Liste der Dinge, die meiner Erfahrung nach darüber entscheiden, ob automatisierter Handel tatsächlich überlebt oder ob er ein Konto langsam von innen heraus zerstört.

Manche davon sind im Nachhinein offensichtlich. Andere werden es erst, nachdem man einem Bekannten dabei zugesehen hat, wie er einen sechsstelligen Betrag verliert. Ich habe versucht, das Ganze in etwa einer halben Stunde lesbar zu halten, ohne dass dafür besondere Werkzeuge nötig sind.

## SO NUTZEN SIE SIE

Vergeben Sie pro Frage null oder einen Punkt. Null heißt, dass Ihr System das nicht tut, oder dass Sie sich nicht sicher sind. Eins heißt, dass es das tut, und zwar so, dass Sie es beschreiben, im Code zeigen und auf Nachfrage demonstrieren können.

Zählen Sie Ihre Punktzahl am Ende zusammen. Die Bewertungsskala steht am Ende des Dokuments.

Zwei kurze Hinweise, bevor Sie anfangen. Seien Sie ehrlich zu sich selbst; das ist ein Werkzeug für Sie, keine Marketingübung. Wenn Sie bei einer Frage, bei der Sie sich nicht sicher sind, ein „Ja“ hinmogeln, optimieren Sie damit für ein gutes Gefühl heute und für Verluste später.

Und denken Sie daran, was diese Checkliste tatsächlich misst. Sie betrachtet die Automatisierung, nicht die Strategie selbst. Eine perfekte Punktzahl macht aus einer schlechten Strategie keine profitable. Was sie aber bewirkt: Wenn Ihre Strategie funktioniert, sorgt die Automatisierung um sie herum nicht dafür, dass die Gewinne leise wieder verloren gehen.

## FRAGE 01 · TAGESVERLUSTLIMIT

## Hat Ihr Bot ein hartes Tagesverlustlimit, das er selbst nicht außer Kraft setzen kann?

### WARUM ES WICHTIG IST

Der zuverlässigste Weg, auf dem Automatisierung ein Konto auslöscht, besteht darin, an einem schlechten Tag einfach weiterzuhandeln. Die meisten Strategien haben pro Jahr eine Handvoll Ausreißer-Tage, an denen die Bedingungen nicht zu dem passen, wofür die Strategie ursprünglich gebaut wurde. Ohne ein hartes Limit wendet der Bot weiterhin normale Logik auf abnormale Bedingungen an, und die Verluste laufen schneller auf, als ein Mensch sie zulassen würde.

### WIE ES RICHTIG AUSSIEHT

Ein konfigurierbarer Schwellenwert (in Dollar oder als Prozentsatz des Konto-Eigenkapitals), der vor jeder Ordereinreichung geprüft wird. Wenn das Limit überschritten wird, hält der Bot an. Von dort hängt das Verhalten davon ab, was vorher definiert wurde: Offene Orders können storniert werden, Positionen können geschlossen oder gehalten werden, und der Trader bekommt eine Benachrichtigung. Der Stopp bleibt bestehen, bis er manuell zurückgesetzt wird oder bis zur nächsten Handelssitzung.

### TYPISCHER FEHLER

*Das Limit existiert als Variable in der Strategie-Datei, wird aber an der Order-Submission-Schicht nie tatsächlich durchgesetzt. Der Bot rast in einem schnellen Markt darüber hinweg, genau dann, wenn es darauf ankommt.*

## FRAGE 02 · POSITIONSGRÖSSEN-CAP

## Gibt es einen harten Cap auf die Positionsgröße, pro Trade und pro Symbol?

### WARUM ES WICHTIG IST

Strategien dimensionieren Positionen oft dynamisch, auf Basis von Volatilität oder Konto-Eigenkapital. Das funktioniert, bis sich ein Input merkwürdig verhält. Ein Volatilitätswert, der auf null zurückfällt (etwa weil im Datenfeed eine Lücke entstand), kann aus einer dynamischen Formel eine enorme Lot-Größe erzeugen, und die Börse füllt sie bereitwillig.

Ein einfacher absoluter Cap, der nach der eigenen Sizing-Logik der Strategie greift, fängt all das ab. Der Cap soll die Sizing-Logik nicht ersetzen. Er ist dafür da, dass wenn diese Logik aus welchem Grund auch immer versagt, der Schaden begrenzt bleibt.

### WIE ES RICHTIG AUSSIEHT

Zwei Obergrenzen. Pro Trade: nie mehr als X Einheiten in einer einzelnen Order. Pro Symbol: nie mehr als Y Einheiten an Gesamtexposure auf einem einzelnen Instrument, unabhängig davon, wie viele Orders nötig waren, um die Position aufzubauen. Beides wird bei der Submission geprüft, nicht nur in der Strategie-Logik. Konfigurierbar pro Instrument, weil das, was für ein Major-Paar vernünftig ist, für einen Small-Cap-Altcoin rücksichtslos wäre.

### TYPISCHER FEHLER

*Der Cap existiert auf Strategie-Ebene, berücksichtigt aber keine Teilausführungen, Retries oder Wiedereinstiege nach einem Stop-out. Beim dritten Wiedereinstieg liegt die tatsächliche Exposure beim Vielfachen des angegebenen Limits.*

## FRAGE 03 · WITHDRAWAL-API

## Ist der API-Schlüssel explizit ohne Withdrawal-Berechtigung eingerichtet?

### WARUM ES WICHTIG IST

Ein Trading-Bot hat keinen legitimen Grund, Mittel abzuheben. Keinen. Falls das System auf irgendeine Weise kompromittiert wird, sei es durch einen geleakten Schlüssel, einen Angreifer auf Ihrem Rechner oder ein böses Library-Update, ist das einzige, was zwischen dem Angreifer und Ihrem Geld steht, die Berechtigung auf diesem Schlüssel.

Deaktivierter Withdrawal bedeutet, dass der schlimmste Fall ungewollte Positionen sind, was sich beheben lässt. Aktivierter Withdrawal bedeutet, dass die Mittel innerhalb von Minuten weg sind.

### WIE ES RICHTIG AUSSIEHT

Jeder von der Automatisierung verwendete Schlüssel wird mit deaktiviertem Withdrawal auf Ebene der Börse angelegt. IP-allowlisted, sofern die Börse das unterstützt. Rotiert nach einem festen Zeitplan, wobei 90 Tage ein vernünftiger Standardwert sind. Aufbewahrt in einem ordentlichen Secret Manager und nicht in einer Konfigurationsdatei oder einer Chatnachricht.

### TYPISCHER FEHLER

*Der ursprüngliche Schlüssel wurde „für alle Fälle“ mit aktiviertem Withdrawal angelegt und lebt jetzt in einer Dotfile auf Ihrem Laptop, dem Cloud-Server und in drei verwaisten Projektordnern.*

## FRAGE 04 · DOPPELTE ORDERS

## Erkennt der Bot nach einem Reconnect korrekt eine Order, die er zwar gesendet hat, für die er aber keine Bestätigung bekam?

### WARUM ES WICHTIG IST

Das ist die Fehlerart, die ich im echten Leben am häufigsten sehe. Der Bot sendet eine Order. Die Verbindung bricht ab, bevor die Antwort zurückkommt. Der Bot verbindet sich neu, fragt offene Orders ab, und die Order ist nicht da, weil sie bereits gefüllt wurde. Der Bot interpretiert „keine offene Order“ als „keine Order wurde gesendet“ und schickt eine weitere. Der Trader landet in der doppelten Position, mit Exit-Logik, die für eine einzelne Einheit geschrieben wurde.

Verbindungsabbrüche sind nicht selten. Sie passieren unter normalen Bedingungen alle paar Stunden, häufiger über eine Privatleitung oder einen günstigen VPS. Jeder Bot, der länger als ein paar Wochen läuft, wird das erleben.

### WIE ES RICHTIG AUSSIEHT

Jede Order trägt eine deterministische `client_order_id`, mit der die Börse Duplikate serverseitig ablehnen kann. Beim Reconnect fragt der Bot die seit dem letzten bekannten Zeitstempel gefüllten Orders ab (nicht nur offene Orders) und gleicht den erwarteten Zustand mit dem tatsächlichen Zustand der Börse ab. Wo Unsicherheit besteht, pausiert der Bot für 30 bis 60 Sekunden, bevor er irgendetwas Neues platziert.

### TYPISCHER FEHLER

*Der Bot vertraut beim Reconnect dem, was er in „offene Orders“ sieht, ohne Abgleich mit der Fill-Historie. Der erste Netzwerkaussetzer verdoppelt die Position still und leise.*

## FRAGE 05 · VERALTETE SIGNALE

## Verwirft der Bot ein Signal, das zu spät ankam, um noch sinnvoll zu sein?

### WARUM ES WICHTIG IST

Handelssignale sind nur innerhalb eines Zeitfensters gültig. Ein TradingView-Alert, der sagt „kaufe zur Eröffnung der 1-Stunden-Kerze“, ist zur Eröffnung dieser Kerze relevant. Drei Stunden später, an einer anderen Kerze, in einer anderen Preislage, ist dasselbe Signal nur noch ein zufälliger Trade.

Webhook-Verzögerungen passieren laufend. TradingView selbst hatte in volatilen Phasen mehrminütige Rückstände. Cloud-Funktionen haben Cold-Starts. Notification-Services stauen sich. Ein Bot, der jedes Signal verarbeitet, unabhängig vom Alter, wird irgendwann einen Trade nach Anweisungen von vor zwanzig Minuten ausführen.

### WIE ES RICHTIG AUSSIEHT

Jedes Signal trägt einen Zeitstempel von der Quelle. Der Bot prüft das Alter des Signals in dem Moment, in dem er handeln will. Wenn das Signal älter ist als ein konfigurierter Schwellenwert (typischerweise Sekunden für Intraday, Minuten für Swing-Setups), wird es geloggt, ein Alert wird ausgelöst und das Signal verworfen.

### TYPISCHER FEHLER

*Der Bot verarbeitet Signale in der Eingangsreihenfolge ohne Altersprüfung. Ein 20-minütiger Rückstand nach einem Webhook-Ausfall produziert 20 Minuten veraltete Trades, sobald die Warteschlange abgearbeitet wird.*

## FRAGE 06 · KILL SWITCH

## Können Sie das System in unter fünf Sekunden vom Telefon aus stoppen, ohne sich bei der Börse einloggen zu müssen?

### WARUM ES WICHTIG IST

Wenn etwas mit einem Live-Bot schief läuft, zählt jede Sekunde. Wenn der einzige Weg zum Stopp darin besteht, sich per SSH auf einen Server zu verbinden und einen Prozess zu beenden, oder sich bei der Börse einzuloggen und manuell anzufangen zu stornieren, wird der Trader entweder erstarren oder den falschen Zug machen. Menschen treffen in den ersten dreißig Sekunden, in denen sie ihrem Konto beim unerwarteten Verhalten zusehen, keine guten Entscheidungen.

Der Kill Switch muss eine einzelne bewusste Aktion sein, von überall aus erreichbar, wo Sie gerade sind.

### WIE ES RICHTIG AUSSIEHT

Eine einfache Schnittstelle, die sich vom Telefon aus in unter fünf Sekunden auslösen lässt. Ein Telegram-Bot-Befehl, ein Knopf auf einem Dashboard, ein SMS-Code, alles, was nicht voraussetzt, in die Börsen-UI zu gelangen. Die Kill-Aktion wird mit Zeitstempel geloggt. Nach der Auslösung kann der Bot erst wieder anlaufen, wenn ein expliziter manueller Neustart erfolgt, idealerweise mit einer Bestätigung, dass der Trader den Zustand geprüft hat.

### TYPISCHER FEHLER

*Der „Kill Switch“ verlangt, dass man sich in eine Cloud-Konsole einloggt, den richtigen Service findet und einen Container stoppt. Der Trader hantiert mit Zwei-Faktor-Authentifizierung, während die Position gegen ihn läuft.*

## FRAGE 07 · PAPER-MODUS

## Läuft der Bot im Paper-Modus über denselben Code-Pfad wie im Live-Betrieb?

### WARUM ES WICHTIG IST

Ein häufiges Muster ist, den Paper-Modus als separaten Branch oder als anderes Skript zu führen. Mit der Zeit driftet die Paper-Version aus dem Sync mit dem Live-Code. Neue Features werden im Live-Code ergänzt und im Paper-Modus vergessen. Bug Fixes werden an einer Stelle angewendet und an der anderen nicht.

Wenn Sie dann eine Änderung im Paper validieren wollen, testen Sie ein System, das dem tatsächlich Handelnden nicht mehr ähnelt. Ein falsches Sicherheitsgefühl ist schlechter als gar kein Paper-Modus.

### WIE ES RICHTIG AUSSIEHT

Alles oberhalb der Ausführungsschicht ist zwischen Paper und Live geteilt: die Risk Engine, die Strategie-Logik, der Order Builder, die Reconciliation. Das Einzige, was sich tatsächlich unterscheidet, ist der Adapter am äußersten Rand. Im Paper-Modus spricht er mit einem Simulator (Testnet, ein interner Simulator oder replayte Live-Daten). Im Live-Modus spricht er mit der echten Börse. Die Umschaltung zwischen beidem ist ein Konfigurationsflag, keine Codeänderung.

### TYPISCHER FEHLER

*Paper und Live sind zwei separate Skripte. Das letzte Mal, dass jemand die Paper-Version angeschaut hat, war vor sechs Monaten, und sie kompiliert nicht einmal mehr gegen das aktuelle Datenschema.*

## FRAGE 08 · ENTSCHEIDUNGS-LOGS

## Wird jede Entscheidung, die der Bot trifft, mit Zeitstempel festgehalten?

### WARUM ES WICHTIG IST

Wenn ein Bot etwas Unerwartetes tut, ist die erste Frage immer „warum“. Die einzige Möglichkeit, das zu beantworten, sind Logs, und die Logs müssen mehr abdecken als nur Orders und Fills. Jedes Signal, das eingegangen ist, jeder Filter, der dagegen lief, jede Risikoprüfung, die durchging oder fehlschlug, jeder Grund, warum ein Trade übersprungen wurde: all das muss dokumentiert sein. Ohne das ist eine Post-Incident-Analyse nur Raten.

Logs erkennen auch langsamen Drift. Ein Filter, der sich still zu eng eingestellt hat. Schleichend steigende Slippage. Allmählich fallende Win-Rate. Nichts davon zeigt sich am täglichen P&L, aber alles davon zählt.

### WIE ES RICHTIG AUSSIEHT

Strukturiertes Logging (JSON oder ähnlich) für jedes signifikante Ereignis. UTC-Zeitstempel mit Millisekunden-Präzision. Klare Eventtypen: signal\_received, signal\_rejected, order\_submitted, order\_filled, risk\_breach, kill\_activated. Mindestens 90 Tage aufbewahrt. Abfragbar, ohne durch Gigabytes Text greppen zu müssen. Schon eine einfache Postgres-Tabelle mit ordentlichen Indizes leistet das.

### TYPISCHER FEHLER

*Der Bot schreibt in eine Text-Logdatei auf irgendeinem Server. Niemand schaut hinein, bis etwas kaputt geht. Bis dahin hat die Datei rotiert, und die relevanten Einträge sind weg.*

## FRAGE 09 · API-SCHLÜSSEL-HYGIENE

## Sind die API-Schlüssel verschlüsselt gespeichert, von der Codebase getrennt und nach einem festen Zeitplan rotiert?

### WARUM ES WICHTIG IST

API-Schlüssel sind Zugangsdaten zu Ihrem Geld. Die meisten Leaks, von denen ich gehört habe, waren keine ausgeklügelten Angriffe. Es waren Schlüssel, die in ein öffentliches Git-Repository committet wurden, in einen Chat eingefügt, um Hilfe beim Debugging zu bekommen, in einem Screenshot beim Support hochgeladen oder in einer Dotfile gespeichert, die in einen Cloud-Drive synchronisiert wurde.

Rotation ist wichtig, weil eine Kompromittierung nicht immer bekannt ist. Ein geleakter Schlüssel, der monatelang ungenutzt herumliegt, ist harmlos. Derselbe Schlüssel in den falschen Händen an einem volatilen Tag kostet viel. Rotation begrenzt das Zeitfenster, in dem ein Leak überhaupt ausgenutzt werden kann.

### WIE ES RICHTIG AUSSIEHT

Schlüssel leben in einem dedizierten Secret Store: einem Managed Service, einem selbstgehosteten Vault oder verschlüsselten Umgebungsvariablen, die von der Hosting-Plattform bereitgestellt werden. Nie in der Codebase. Nie im Klartext auf der Platte. Rotiert nach einem festen Zeitplan (90 Tage sind sinnvoll), und der Rotationsprozess wird vorher getestet, damit er den Bot nicht zerlegt, wenn er tatsächlich greift.

### TYPISCHER FEHLER

*Der Trader weiß, dass die Schlüssel „irgendwo sicher“ liegen, kann aber nicht zeigen wo, und die letzte Rotation war an dem Tag, an dem der Bot zum ersten Mal deployed wurde.*

## FRAGE 10 · INCIDENT-PROTOKOLL

## Gibt es ein schriftliches Vorgehen für den Fall, dass die Börse eine Order ablehnt oder einen Fill nicht bestätigt?

### WARUM ES WICHTIG IST

Börsen sind keine vollständig zuverlässigen Systeme. Sie lehnen Orders aus Gründen ab, die nicht immer Sinn ergeben. Manchmal nehmen sie eine Order an und melden sie dann als abgelehnt zurück. Gelegentlich verlieren sie den Überblick über Fills und brauchen eine Reconciliation. Ein Bot, der annimmt, die Börse antwortet immer korrekt, wird früher oder später in einem Zustand landen, in dem seine interne Sicht nicht mehr mit der Sicht der Börse übereinstimmt.

Das Incident-Protokoll ist die Antwort auf die Frage „was tut der Bot, wenn Realität und Erwartung auseinandergehen“.

### WIE ES RICHTIG AUSSIEHT

Eine dokumentierte Liste von Fehlerszenarien, jedes mit einer definierten Reaktion. Für jeden Fehlertyp gibt es eine klare Antwort darauf, wann ein Retry sicher ist, wann der Trader benachrichtigt werden muss anstatt es still zu behandeln, und wann das gesamte System pausieren soll, bis ein Mensch bestätigt, in welchem Zustand die Dinge tatsächlich sind. Das Protokoll ist schriftlich, nicht nur in Code implementiert, damit der Trader es lesen und an veränderte Bedingungen anpassen kann.

### TYPISCHER FEHLER

*Die Fehlerbehandlung ist das, was dem Entwickler gerade einfiel zu behandeln. Unerwartete Fehler bringen den Prozess zum Absturz, er wird auto-neugestartet, und der Bot nimmt den Handel wieder auf, ohne dass jemand bemerkt, dass der Zustand zurückgesetzt wurde.*

## FRAGE 11 · STUFENWEISER START

## Wurde dieses System in Stufen gestartet, oder war Tag eins bereits ein Deployment in voller Größe?

### WARUM ES WICHTIG IST

Strategien sehen in Backtests besser aus, als sie live performen. Das ist nicht umstritten. Slippage ist in der Realität größer. Fills sind langsamer. Manche Setups triggern nie tatsächlich, weil die Daten, die der Backtest verwendet hat, sauberer waren als der Live-Feed. Jede Automatisierung hat Bugs, die sich erst zeigen, wenn sie gegen tatsächliches Börsenverhalten läuft.

Ein stufenweiser Start lässt Sie diese Probleme mit kleinem Geld finden statt mit voller Größe. Die übliche Reihenfolge ist erst Paper, dann Shadow-Modus (Live-Daten, aber keine realen Orders), dann Micro-Live in der kleinstmöglichen Größe, die noch etwas aussagt, und erst dann schrittweise größer, wenn das Vertrauen in das System tatsächlich wächst.

### WIE ES RICHTIG AUSSIEHT

Ein vorab vereinbarter Launch-Plan mit expliziter Dauer und Erfolgskriterien pro Stufe. Die Positionsgröße in Micro-Live ist klein genug, dass ein Totalverlust des Testkapitals nicht ins Gewicht fällt. Die Beförderung von einer Stufe zur nächsten verlangt ein positives Review, nicht nur verstrichene Zeit. Der Plan ist schriftlich, bevor der Bot live geht.

### TYPISCHER FEHLER

*Der Bot wird „im Paper getestet“ für ein paar Tage, dann an ein Live-Konto in normaler Größe angeschlossen, weil die Paper-Ergebnisse „gut aussahen“. Der erste echte Bug taucht an einer echten Position auf.*

## FRAGE 12 · DASHBOARD

## Gibt es ein Dashboard, getrennt von der Börsen-UI, in das Sie regelmäßig hineinschauen?

### WARUM ES WICHTIG IST

Die meisten Trader prüfen ihren Bot, indem sie sich bei der Börse einloggen und Positionen anschauen. Das sagt Ihnen, was die Börse glaubt, das passiert ist, nicht was der Bot glaubt, das passiert ist. Das Auseinanderdriften dieser beiden Sichten ist genau das, was früh erkannt werden sollte, und der einzige Weg, es zu erkennen, ist, sich die Sicht des Bots unabhängig anzuschauen.

Das Dashboard ist auch der Ort, an dem langsamer Drift sichtbar wird. Eine über Wochen fallende Win-Rate. Schleichend steigende Slippage. Höhere Ablehnungsraten bei Signalen, weil ein Filter zu eng geworden ist. Nichts davon zeigt sich am täglichen P&L, aber alles davon zählt.

### WIE ES RICHTIG AUSSIEHT

Eine einfache Web-Oberfläche (sie muss nicht hübsch sein), die den aktuellen Zustand zeigt: aktive Positionen, jüngste Fills, jüngste Signale, jüngste Ablehnungen, aktuelle Risiko-Auslastung gegen die Limits, letzter Kill-Switch-Status. Historische Sicht für mindestens 30 Tage. Vom Telefon aus erreichbar. Nach einem festen Zeitplan geprüft, mindestens täglich.

### TYPISCHER FEHLER

*Es gibt kein Dashboard. Der Trader prüft gelegentlich die Börsen-App und vertraut darauf, dass der Bot in Ordnung ist, weil die Equity-Kurve „in etwa flach“ verläuft.*

## FRAGE 13 · SLIPPAGE

## Trackt der Bot Slippage bei jedem Fill und pausiert er, wenn die Slippage einen Schwellenwert überschreitet?

### WARUM ES WICHTIG IST

Slippage ist die Differenz zwischen dem Preis, den der Bot erwartet hat, und dem Preis, den er tatsächlich bekommen hat. Unter normalen Bedingungen ist Slippage klein und vorhersehbar. Unter abnormalen Bedingungen kann Slippage enorm werden, manchmal um Größenordnungen schlechter, als der Backtest angenommen hat. Die abnormalen Bedingungen variieren: schnelle Märkte, dünne Liquidität, Börsenausfälle, regulatorische Nachrichten mitten in der Session.

Ein Bot, der Slippage ignoriert, handelt fröhlich weiter durch diese Bedingungen hindurch, bekommt zu immer schlechteren Preisen Fills, bis der Edge der Strategie von den Ausführungskosten aufgeessen ist.

### WIE ES RICHTIG AUSSIEHT

Slippage wird für jeden Fill in Basispunkten gegenüber dem erwarteten Preis gemessen. Gleitender Durchschnitt über ein kurzes Fenster. Wenn der Durchschnitt einen Schwellenwert überschreitet (alles über dem 5-fachen Normalwert ist verdächtig), pausiert der Bot den Handel und schickt einen Alert. Der Schwellenwert ist pro Instrument konfigurierbar.

### TYPISCHER FEHLER

*Slippage wird gar nicht gemessen. Der Backtest der Strategie ging von einem Basispunkt Slippage aus. Live-Bedingungen während eines Stress-Events produzieren fünfzig, und der Bot handelt weiter.*

## FRAGE 14 · AVERAGING-CAP

## Wird die maximale Anzahl an Averaging- oder DCA-Schritten im Code erzwungen, nicht nur im Strategie-Dokument?

### WARUM ES WICHTIG IST

Viele Strategien verwenden Averaging, also das Nachkaufen in eine Position, während diese sich gegen den ursprünglichen Einstieg bewegt. Das funktioniert in einem normalen Markt. Es produziert auch einige der schlimmsten Blow-ups im Retail-Trading, weil die Vorstellung der Strategie davon, „wie oft kann ich nachkaufen“, typischerweise größer ist, als das Konto während einer ernsten Bewegung tatsächlich tragen kann.

Ein Strategie-Dokument mag sagen „bis zu fünfmal nachkaufen“. Der Code, sofern ihm nicht ausdrücklich etwas anderes gesagt wird, kauft so oft nach, wie das Signal feuert. Während einer schnellen adversen Bewegung können das zehn oder zwanzig Male sein.

### WIE ES RICHTIG AUSSIEHT

Ein harter Zähler in der Ausführungsschicht, der die Averaging-Schritte pro Position verfolgt. Nach dem konfigurierten Limit werden keine weiteren Averaging-Orders akzeptiert, unabhängig davon, was das Signal sagt. Der Zähler wird nur zurückgesetzt, wenn die Position vollständig geschlossen ist, nicht wenn sie nur teilweise reduziert wurde. Das Limit ist konservativ, niedriger als das, was die Strategie in einem normalen Markt „könnte“, weil der Cap für abnormale Märkte existiert.

### TYPISCHER FEHLER

*Das Averaging-Limit lebt in der Strategie-Datei als Variable, die die Strategie respektieren soll. Während einer schnellen Bewegung bekommt die Strategie-Logik keine Gelegenheit, das zu prüfen, und der Bot kauft weiter nach in einen freien Fall.*

## Bewertung.

Addieren Sie Ihre Nullen und Einsen über die 14 Fragen. Das Ergebnis sagt Ihnen, wo Ihr System steht.

**0–5****KRITISCHES RISIKO**

Das System ist einen schlechten Tag entfernt von einem ernsten Verlust, der nichts mit der Strategie zu tun hat. Empfehlung: Live-Handel einstellen, bis die Lücken geschlossen sind.

**6–9****ERHEBLICHE LÜCKEN**

Das System wird die meiste Zeit funktionieren, aber die nicht abgedeckten Fehlerarten sind diejenigen, die den größten Schaden anrichten. Ein fokussiertes Projekt über ein bis zwei Wochen lohnt sich, um die ungeprüften Punkte zu schließen.

**10–12****SOLIDE**

Die großen Risiken sind abgedeckt. Verbleibende Punkte betreffen typischerweise die Qualität des Monitorings, operative Disziplin oder Grenzfälle, die der Trader bewusst akzeptiert hat.

**13–14****INSTITUTIONAL-GRADE**

Das System ist auf dem Standard gebaut, wie ein ordentlich geführter Trading-Betrieb aussehen sollte. Die verbleibende Sorge ist meist Drift über die Zeit: operative Praktiken, die beim Start gut waren, schleifen sich ab, wenn das System jahrelang läuft.

## Wenn Sie eine zweite Meinung wollen.

Wenn Sie unter 10 gepunktet haben und eine zweite Meinung dazu wollen, welche Lücken für Ihr konkretes Setup am wichtigsten sind, schreiben Sie mir. Ich biete bezahlte Hardening-Audits auf bestehender Automatisierung an. Das Format ist ein 30-Minuten-Gespräch, gefolgt von einem schriftlichen Review gegen diese Checkliste und gegen Ihren tatsächlichen Code und Ihre Konfiguration. Das Deliverable ist eine priorisierte Liste: was jetzt zu beheben ist, was so in Ordnung ist, was es nicht wert ist, angefasst zu werden.

Wenn Sie über 10 gepunktet haben und darüber nachdenken, das auszuweiten, was Ihre Automatisierung abdeckt (neue Märkte, zusätzliche Strategien, schärfere Risikokontrollen), ist das ein anderes Gespräch, das ich aber auch gerne führe.

### **Pavlo Filianov**

Gründer, FAMBF — Framework for Automated, Mandate-Backed Finance

25 Jahre an den Märkten, die ersten 13 teilweise durch ein lizenziertes Brokerhaus, das ich in der Ukraine gegründet habe, und das letzte Jahrzehnt überwiegend in Krypto. Ich baue Automatisierung für eigenverantwortliche Trader, die wollen, dass ihre Regeln tatsächlich auf ihrem eigenen Konto laufen.

- [fambf.com](https://fambf.com)
- [pavlo@fambf.com](mailto:pavlo@fambf.com)
- 30-Minuten-Gespräch buchen
- [LinkedIn](#)