# **Solo Founder: Week-by-Week Execution Checklist**

**Owner**: Assaf  
 **Timeline**: 12 weeks to first paying customer  
 **Updated**: Week by week progress tracking

## **Phase 1: Compliance-Proven Core (Weeks 1-4)**

### **Week 1: Foundation Building**

**Goal**: Productize your MVP learnings into reliable core components

#### **Monday-Tuesday: Compliance-First Driver**

* [ ] **Morning**: Set up clean project structure
* [ ] **Day 1**: Build claude\_driver.py with your proven MVP approach
* [ ] **Day 1**: Implement compliance\_checker.py with zero-inference validation
* [ ] **Day 2**: Create audit\_logger.py for complete request tracing
* [ ] **Day 2 Evening**: Test driver with simple prompts, verify compliance catching

**Success Check**: Driver blocks synthetic data generation and logs everything

#### **Wednesday-Thursday: PDF + Span Preservation**

* [ ] **Day 3**: Build pdf\_processor.py using PyPDF2 (you know this works)
* [ ] **Day 3**: Create span\_tracker.py for character-level position tracking
* [ ] **Day 4**: Test PDF processing with FBI/CISA sample reports
* [ ] **Day 4**: Verify span positions are accurate for citation generation

**Success Check**: Can extract text and cite exact positions for every element

#### **Friday: Integration + Testing**

* [ ] **Morning**: Connect driver + PDF processor into working pipeline
* [ ] **Afternoon**: Test end-to-end with 3-4 different intelligence reports
* [ ] **Evening**: Fix any integration bugs, document what you learned

**Week 1 Demo**: PDF in → Text extraction with spans → Audit trail out

### **Week 2: Zero-Inference Extraction**

**Goal**: Build the core differentiator that makes Safety Sigma unique

#### **Monday-Wednesday: IR Extractor**

* [ ] **Day 1**: Build ir\_extractor.py main orchestrator
* [ ] **Day 1**: Implement your refined zero\_inference\_prompt.py from MVP learnings
* [ ] **Day 2**: Create citation\_engine.py to link every output to source spans
* [ ] **Day 2**: Test extraction on FBI romance scam report (known good example)
* [ ] **Day 3**: Build basic ThreatPattern data model (simplified from complex design)
* [ ] **Day 3**: Test extraction on CISA infrastructure alert

**Success Check**: Extracts structured data with zero synthetic elements

#### **Thursday-Friday: IR Schema + Testing**

* [ ] **Day 4**: Define core IR schema (temporal, financial, infrastructure, behavioral)
* [ ] **Day 4**: Test schema with different report types
* [ ] **Day 5**: Integration testing: PDF → Extraction → Structured IR
* [ ] **Day 5**: Validate all extracted elements have source citations

**Week 2 Demo**: Intelligence report → Structured threat patterns with citations

### **Week 3: Demo-Ready Package**

**Goal**: Make it customer-presentable and professional

#### **Monday-Tuesday: Simple Rule Generation**

* [ ] **Day 1**: Build SQL query generation from IR patterns
* [ ] **Day 1**: Test SQL output on sample threat patterns
* [ ] **Day 2**: Build Python script generation from IR patterns
* [ ] **Day 2**: Test Python output execution

**Success Check**: Generated rules are syntactically correct and runnable

#### **Wednesday: Basic Validation**

* [ ] **Day 3 Morning**: Implement hallucination detection (your key differentiator)
* [ ] **Day 3 Afternoon**: Build citation verification system
* [ ] **Day 3 Evening**: Test validation catches compliance violations

**Success Check**: Validation blocks any synthetic data or invalid citations

#### **Thursday-Friday: Demo Polish**

* [ ] **Day 4**: Build clean Streamlit interface for demonstrations
* [ ] **Day 4**: Create professional output formatting (HTML + JSON)
* [ ] **Day 5 Morning**: Set up sample FBI report processing demo
* [ ] **Day 5 Afternoon**: Practice 5-minute demo presentation
* [ ] **Day 5 Evening**: Record demo video for async sharing

**Week 3 Demo**: Full pipeline with professional interface

### **Week 4: Customer Demo Week**

**Goal**: Get customer feedback and validate willingness to pay

#### **Monday: Demo Preparation**

* [ ] **Morning**: Finalize demo environment and backup plans
* [ ] **Afternoon**: Prepare demo script and FAQ responses
* [ ] **Evening**: Send demo invites to 10+ prospects

#### **Tuesday-Friday: Customer Demos**

* [ ] **Target**: 2-3 demos per day, 8-10 total for the week
* [ ] **Focus Questions**:
  + "Would this solve a real problem for you?"
  + "What would you pay monthly for this capability?"
  + "What's missing to make this useful for your team?"
  + "When would you want to start using this?"

#### **Demo Feedback Tracking**

* [ ] **After each demo**: Document feedback in structured format
* [ ] **Weekly summary**: Identify top 3 requested features
* [ ] **Decision point**: Do ≥3 customers say "I would pay for this"?

**Week 4 Success**: Clear customer validation and feature requests

## **Phase 2: Customer-Requested Features (Weeks 5-7)**

### **Week 5: #1 Customer Request**

**Goal**: Build the most requested feature from Week 4 demos

#### **Monday: Feature Planning**

* [ ] **Morning**: Analyze Week 4 feedback, identify #1 request
* [ ] **Afternoon**: Break down feature into 4-day implementation plan
* [ ] **Evening**: Set up development environment for new feature

#### **Tuesday-Friday: Feature Implementation**

* [ ] **Day 2-3**: Core feature development
* [ ] **Day 4-5**: Integration with existing system + testing

**Feature likely to be one of**:

* Specialized threat type handling
* New export format (Unit21, Sift, etc.)
* Enhanced audit trail visualization
* Side-by-side comparison with manual process

**Week 5 Demo**: Show enhanced system to 3-4 customers who requested this

### **Week 6: #2 Customer Request**

**Goal**: Build the second most requested feature

#### **Monday: Feature Planning**

* [ ] **Morning**: Plan #2 customer request implementation
* [ ] **Afternoon**: Verify integration approach with existing features

#### **Tuesday-Friday: Feature Implementation**

* [ ] **Day 2-4**: Core development
* [ ] **Day 5**: Testing + integration

**Week 6 Demo**: Show both new features to customer group

### **Week 7: Polish + Validation**

**Goal**: Refine based on feedback and validate pricing

#### **Monday-Tuesday: Polish**

* [ ] **Day 1**: Fix bugs found in Week 5-6 demos
* [ ] **Day 2**: Improve user experience based on feedback

#### **Wednesday-Friday: Customer Validation**

* [ ] **Day 3-5**: Demo improved version to same customers
* [ ] **Focus Questions**:
  + "Is this ready for your team to use?"
  + "What would you pay monthly for this?"
  + "What would prevent you from buying this?"
  + "When could you start paying for this?"

**Week 7 Success**: ≥3 customers say "When can we buy this?"

## **Phase 3: Revenue-Ready System (Weeks 8-12)**

### **Week 8-9: Production Essentials**

**Goal**: Build minimum viable production system

#### **Week 8: Reliability**

* [ ] **Day 1-2**: Implement proper error handling and user-friendly error messages
* [ ] **Day 3**: Add rate limiting to prevent API quota issues
* [ ] **Day 4**: Basic security (API key management, input validation)
* [ ] **Day 5**: Simple monitoring (health checks, basic metrics)

#### **Week 9: Deployment**

* [ ] **Day 1-2**: Create Docker container for easy deployment
* [ ] **Day 3**: Set up deployment on Railway/Fly.io
* [ ] **Day 4**: Test deployment with customer data
* [ ] **Day 5**: Create simple deployment documentation

**Week 9 Success**: System runs reliably in production environment

### **Week 10: Customer Onboarding**

**Goal**: Make it easy for customers to get started

#### **Monday-Tuesday: Documentation**

* [ ] **Day 1**: Write clear setup guide and API documentation
* [ ] **Day 2**: Create troubleshooting guide for common issues

#### **Wednesday-Thursday: Support Infrastructure**

* [ ] **Day 3**: Set up customer support system (email + simple chat)
* [ ] **Day 4**: Create customer feedback tracking system

#### **Friday: Onboarding Testing**

* [ ] **Day 5**: Test complete customer onboarding flow
* [ ] **Evening**: Document any friction points

**Week 10 Success**: Smooth customer onboarding experience

### **Week 11: Pricing + Sales**

**Goal**: Enable customers to actually pay you

#### **Monday-Tuesday: Pricing Model**

* [ ] **Day 1**: Finalize pricing based on customer feedback (per document, monthly, etc.)
* [ ] **Day 2**: Set up Stripe for payment processing

#### **Wednesday-Thursday: Sales Process**

* [ ] **Day 3**: Create simple checkout flow and customer dashboard
* [ ] **Day 4**: Set up basic terms of service and privacy policy

#### **Friday: Sales Testing**

* [ ] **Day 5**: Test complete purchase flow end-to-end
* [ ] **Evening**: Prepare for first customer sales calls

**Week 11 Success**: Customers can buy and pay for the product