

Project #3: Spec → Ticket Railgun

From PRD to Jira in 15 Minutes

Executive Summary

Project Name: SpecFlow (working title)

Tagline: "Kill spec drift, ship faster"

RICE Score: 144 (3rd highest priority)

Target Market: PMs and tech leads at seed-Series A companies

Build Time: 3 weeks MVP + 48 hours Custom GPT

Revenue Model: €199/month team + €399/month pro

Problem Statement

Product teams waste **4+ hours per PRD** converting specs into Jira tickets:

- **Spec drift:** Requirements get stale between writing and implementation
- **Context loss:** Engineers implement outdated or incomplete requirements
- **Manual copy-paste:** PMs rewrite the same stories in multiple formats
- **Ambiguity:** Missing acceptance criteria leads to rework
- **No validation:** No checks for completeness or consistency

Market Size: 500+ seed-A startups with product teams. PM tools market growing rapidly. Existing tools ([Dume.ai](#) at \$49/month, ChatPRD) validate demand.

Solution Overview

SpecFlow transforms PRDs into production-ready Jira tickets:

- **Input:** PRD document (Notion, Google Docs, Markdown)
- **Output:** Jira tickets with acceptance criteria, test stubs, priority scoring
- **Intelligence:** Ambiguity detection, scope normalization, dependency mapping
- **Integration:** Jira, Linear, Asana, Azure DevOps

Founder-Market Fit

Why You're Perfect:

- Fractional CTO positioning = deep understanding of PM/eng handoff pain
- Newsletter audience includes product leaders at target companies
- Can differentiate with better Jira integration than existing tools
- Technical expertise (FastAPI, Python, AI/ML) enables rapid build
- Existing client relationships = pilot pipeline ready

Technical Architecture

System Components

1. Custom GPT (Quick Win - 48 hours)

- PRD parser (Notion, Google Docs, Markdown)
- Ambiguity detector (flags vague requirements)
- Ticket draft generator with acceptance criteria
- Definition of Ready scoring

2. Integration Engine (FastAPI)

- Jira API (read/write tickets, custom fields)
- Linear API integration
- Asana integration
- Notion/Google Docs webhook receivers

3. Intelligence Layer

- PRD structure analyzer (sections, features, requirements)
- Scope normalization (standardize formats)
- Acceptance criteria generator (Given/When/Then)
- Test stub creator (unit, integration, e2e templates)
- Priority scoring algorithm

4. Dashboard (HTMX + Tailwind)

- PRD upload and preview
- Ticket generation workflow
- Review and edit interface
- Cycle time analytics
- Team collaboration features

Data Models

```
class PRD(BaseModel):
    prd_id: str
    title: str
    raw_content: str
    parsed_sections: dict
    features: List[Feature]
    metadata: dict
    created_at: datetime

class Feature(BaseModel):
    feature_id: str
    name: str
    description: str
    requirements: List[Requirement]
    acceptance_criteria: List[str]
    test_stubs: List[str]
    priority: int
    complexity: str
    dependencies: List[str]

class JiraTicket(BaseModel):
    ticket_id: str
    project_key: str
    summary: str
    description_html: str
    acceptance_criteria: List[str]
    test_cases: List[str]
    priority: str
    labels: List[str]
    assigned_to: Optional[str]
```

MVP Feature Set (3-Week Build)

Week 1: Core Parser & Generator

- PRD parser (Notion, Markdown, Google Docs)
- Feature extraction and normalization
- Acceptance criteria generator
- Custom GPT prototype

Week 2: Jira Integration

- Jira API integration (create/update tickets)
- Custom field mapping
- Bulk ticket creation workflow
- Ticket preview and editing

Week 3: Intelligence & Polish

- Ambiguity detection algorithm
- Definition of Ready scoring
- Test stub generation
- Dashboard and analytics
- 3 pilot implementations

Quick Win Strategy (48-72 Hours)

Custom GPT "PRD Parser":

1. Input: Upload PRD document or paste text
2. Output: Structured feature list with acceptance criteria
3. Feature: Flag ambiguous requirements for clarification
4. Export: Formatted for Jira import

Launch Plan:

- Substack post: "How to eliminate spec drift"
- LinkedIn demo: "PRD → Jira in 90 seconds"
- Offer free PRD analysis to 10 product teams
- Close 3 early adopters at €99/month (50% discount)

Pricing Strategy

Team Plan: €199/month (5-20 users)

- Unlimited PRDs and tickets
- Jira + Linear integration
- Email support
- Basic analytics

Pro Plan: €399/month (21-50 users)

- All team features
- Custom workflows and templates
- Advanced analytics (cycle time, bottleneck detection)
- Priority support
- API access

Enterprise: Custom pricing

- Unlimited users

- All integrations (Asana, Azure DevOps, etc.)
- SSO and security features
- Dedicated customer success
- Custom integrations

Go-to-Market Strategy

Phase 1: Content Marketing (Week 1-2)

- Substack series: "The PRD → implementation gap"
- LinkedIn posts: "5 signs your spec is drifting"
- Free PRD audit tool (lead magnet)
- Product Hunt teaser page

Phase 2: Pilot Program (Week 3-4)

- Recruit 10 product teams from newsletter
- Offer 50% lifetime discount for pilots
- Success metric: <15 min PRD → tickets
- Document time savings and ROI

Phase 3: Public Launch (Week 5-8)

- Product Hunt launch with case studies
- Demo webinar for newsletter audience
- Partnership with PM communities (Product School, Mind the Product)
- Content: comparison vs Dume.ai, ChatPRD
- Target: 20-30 customers (€3,980-5,970 MRR)

Competitive Analysis

vs Manual Process:

- 93% time savings (4 hours → 15 minutes)
- Standardized ticket quality
- No context loss between PRD and tickets

vs Dume.ai (\$49/month):

- Better Jira integration (bulk creation, custom fields)
- Ambiguity detection (unique feature)

- Test stub generation
- Cycle time analytics

vs ChatPRD:

- Purpose-built for product teams (not generic AI)
- Native integrations (no copy-paste)
- Team collaboration features
- Definition of Ready scoring

Success Metrics (30 Days)

- ✓ 10 paid pilots at €99-199/month (€990-1,990 MRR)
- ✓ Average PRD → tickets time: <15 minutes
- ✓ Ticket quality: >4/5 engineering satisfaction
- ✓ Spec drift reduction: measurable improvement
- ✓ 50+ waitlist signups from content
- ✓ 3-5 case studies with quantified ROI

Claude Code Build Plan

Git Worktree Structure

```
/specflow-mvp/
  /backend/           # FastAPI parser and integrations
  /parser/            # PRD parsing logic (Notion, Docs, MD)
  /integrations/       # Jira, Linear, Asana APIs
  /intelligence/       # AI analysis and scoring
  /dashboard/         # HTMX UI for ticket workflow
  /tests/             # Parser, integration, and e2e tests
```

Claude Prompts

Phase 1 - PRD Parser (Week 1):

```
Prompt: "Create PRD parser supporting Notion, Markdown, Google Docs:
- Extract title, sections, features, requirements
- Normalize structure (handle different formats)
- Identify user stories, acceptance criteria, edge cases
- Flag incomplete or ambiguous sections
Output: structured JSON with features and requirements"
```

Phase 2 - Jira Integration (Week 2):

```
Prompt: "Build Jira integration with FastAPI:
- OAuth authentication flow"
```

```
- Create tickets with custom fields
- Bulk ticket creation (transaction handling)
- Ticket preview before creation
- Error handling and rollback
Include tests for API interaction"
```

Phase 3 - Intelligence Layer (Week 3):

```
Prompt: "Implement PRD intelligence features:
- Ambiguity detection: flag vague language
- Acceptance criteria generator (Given/When/Then)
- Test stub templates (unit, integration, e2e)
- Definition of Ready scoring (0-100 scale)
- Priority recommendation based on impact/effort
Use GPT-4 for natural language analysis"
```

Validation Checklist

Week 1 Validation:

- ☐ Parser handles 3 PRD formats (Notion, Markdown, Docs)
- ☐ Feature extraction accuracy >90%
- ☐ Ambiguity detection identifies 80%+ vague requirements
- ☐ Custom GPT generates quality acceptance criteria

Week 2 Validation:

- ☐ Jira integration creates valid tickets
- ☐ Bulk creation works (10+ tickets simultaneously)
- ☐ Ticket preview shows accurate data
- ☐ Error handling prevents partial creation

Week 3 Validation:

- ☐ 5 pilot teams test end-to-end workflow
- ☐ Average time <15 minutes confirmed
- ☐ Engineering satisfaction >4/5 stars
- ☐ Pricing validated (€199-399 range)

Risk Mitigation

Technical Risks:

- PRD format variations → Support most common formats first
- Jira API changes → Version API client, monitor changelog

- AI accuracy issues → Human review workflow before creation

Market Risks:

- Low adoption → Free trial + ROI calculator
- Competition from [Dume.ai](#) → Emphasize superior integration
- PM workflow resistance → Change management content

Next Steps (This Week)

Day 1: Build Custom GPT prototype

- Test with 3 real PRDs from consulting experience
- Validate acceptance criteria quality
- Identify parsing edge cases

Day 2: Network outreach

- Email 10 PMs from CodeSwiftr clients
- Offer free PRD analysis
- Schedule validation calls

Day 3-5: Core development

- Build PRD parser (Notion + Markdown)
- Jira integration basics
- Ticket generation workflow

Weekend: Pilot testing

- Onboard 2-3 teams for testing
- Measure time savings
- Iterate based on feedback

Long-Term Vision (6-12 Months)

Product Evolution:

- Spec versioning and change tracking
- Real-time collaboration on requirements
- Sprint planning automation
- Backlog health monitoring

Business Evolution:

- €10k MRR from 25-50 teams

- Integration marketplace (Azure DevOps, Monday, etc.)
- API for custom workflows
- White-label for consultancies

Strategic Positioning:

- #1 PRD → implementation tool for seed-A teams
- Active PM community (webinars, templates, best practices)
- Partnership with project management tools
- Thought leadership: spec quality and engineering efficiency

Ready to kill spec drift and help teams ship faster!