PROMPT: PRD v2

The Updated PRD Prompt

Sugartown PRD Template & Guidelines

When to Create a PRD

- New product feature or capability area
- Major technical migration
- Standalone tool or workflow
- Any work requiring >2 weeks of focused effort

Required Structure

1. Meta Header (Always First)

```
---
## Meta
| Field | Value |
|-------|-------|
| **Project ID** | PROJ-### |
| **PRD Version** | vX.Y – [Descriptive Name] |
| **PRD Status** | Draft / Active / Shipped / Deprecated |
| **Owner** | [Name/Role] |
| **Primary Repo** | [GitHub URL] |
| **Scope** | [One-liner description] |
| **Related Standards** | [Links] |
| **Related Gems** | [gem-###] |
| **Supersedes** | [Previous PRD, if any] |
| **Last Reviewed** | YYYY-MM-DD |
---
**Versioning Note:** PRD versions track document evolution. Production releases use
calendar versioning (vYYYY.MM.DD) in CHANGELOG.md.
```

2. Overview

- **Problem statement** - What are we solving?
- **Goals** - What does success look like?
- **Non-goals** - What are we explicitly NOT doing?
- **Success metrics** - How will we measure impact?

3. Context

- **Background** - Why now?
- **User needs** - Who is this for?
- **Technical constraints** - What are we working with?
- **Dependencies** - What do we need from other systems?

4. Proposed Solution

- **High-level approach** - What's the strategy?
- **Architecture** - How does it work?
- **User experience** - What's the interaction model?
- **Technical design** - Key implementation details

5. Phases (Implementation Stages)

```
## Phases

### Phase 0: Incubation (if applicable)
**Scope:** Experimental work, proof of concepts
**Deliverables:** [List]
**Estimated effort:** [X weeks] (estimates in descriptions, NOT phase names)

### Phase 1: Foundation
**Scope:** Core infrastructure, data layer
**Deliverables:** [List]
**Estimated effort:** [X weeks]

### Phase 2: [Descriptive Name]
...
```

Phase Guidelines:

- Use descriptive names, not time ranges ("Foundation" not "Weeks 1-2")
- Phases inform roadmap but aren't 1:1 with sprints
- Can span multiple platform versions
- Time estimates go in descriptions, not phase titles

6. Roadmap (Platform Integration)

```
## Roadmap

| Platform Version | Scope | Status |
|------------------|-------|--------|
| v4.2 | Phase 1 + Phase 2 partial | In Progress |
| v4.3 | Phase 2 completion | Planned |
```

Roadmap Guidelines:

- Show how phases map to platform milestones
- Platform versions are strategic (v4.2, v4.3), releases are tactical (v2025.01.07)
- Avoid date commitments; use status (Planned / In Progress / Shipped)

7. Open Questions & Risks

- **Unresolved decisions** - What do we still need to figure out?
- **Technical risks** - What could go wrong?

- **Dependencies** - What are we waiting on?

8. Appendices (Optional)

- Migration plans
- Technical specifications
- Research findings
- Alternative approaches considered

PRD Versioning

- **Filename:** Stable, no version number (`sugartown_resume_factory_` PRD.md)
- **Meta header:** Semantic PRD version (`PRD Version: v3.0`)
- **Increment:**
  - Major (vX.0): Complete rewrite, new approach, major scope change
  - Minor (vX.Y): Refinements, added sections, updated details

File Naming Convention

- Format: `sugartown_[feature]_` PRD.md
- Examples:
  - `sugartown_cms_` PRD.md (PROJ-001)
  - `sugartown_resume_factory_` PRD.md (PROJ-002)
  - `sugartown_design_system_` PRD.md (PROJ-003)
  - `sugartown_knowledge_graph_` PRD.md (PROJ-004)

Tone & Voice

- **Audience:** Future you, recruiters, developers, hiring managers, collaborators, AI assistants
- **Style:** Product Manager voice - technical and professional
- **Clarity:** Favor specificity over brevity
- **Honesty:** Document risks, unknowns, and constraints openly

Maintenance

- Review PRD when scope changes significantly
- Update Last Reviewed date monthly during active development
- Bump PRD version when structure changes (not just typo fixes)
- Archive PRDs for shipped/deprecated features (don't delete)

```
---

## The Master Reconciliation Table

**What confused us:**

| **Thing We Found** | **What We Thought It Was** | **What It Actually Is** | **Fixed By** |
|--------------------|---------------------------|-------------------------|--------------|
```

`sugartown_2_PRD.md`	PRD v2? Platform v2?	Platform v2.0 reference (legacy)	Rename to `sugartown_cms_PRD.md`, add meta header
`PRD_v3.md`	PRD version	Feature version (Resume Factory v3)	Rename file, separate PRD version from feature version
"PRD Version: 1.5"	Just this one doc's thing	**THE CORRECT APPROACH**	Implement everywhere
PROJ-001	Project version	Project ID (never changes)	Document as permanent identifier
Phase 1 (Weeks 1-2)	Implementation stage	Implementation stage WITH time estimate	Remove time from phase name, add to description
"v2 → v3 Migration"	PRD migration?	Feature migration (Resume Factory evolution)	Clarify in PRD sections

---

The Final PM Word

PM: Okay. So here's what we're doing.

Caucus: Listening.

PM:
1. We're creating TWO docs: Generic version strategy + Sugartown addendum
2. We're updating the PRD prompt with the meta header standard
3. We're cleaning up all four PRDs according to the checklist
4. We're NEVER putting versions in filenames again
5. We're documenting this chaos in a gem

Caucus: A gem about versioning strategy?

PM: A gem about how we discovered we had seven overlapping version systems and lived to tell the tale.

Caucus: Should we version the gem?

PM: *[long pause]*

Caucus: That was a joke.

PM: ...was it though?

Caucus: ...we should probably version the gem.

PM: I'm going to touch grass now.

Next Steps

1. **Create `docs/versioning_strategy.md`** (generic template + Sugartown addendum)
2. **Update `docs/prd_template.md`** (new meta header standard, phase guidelines)
3. **Clean up existing PRDs** (follow checklist above)
4. **Create gem-###** documenting this entire reconciliation process
5. **Update project knowledge** with versioning hierarchy
6. **Never speak of "pink stink" again**

Status: Reconciled (for real this time) ✅
Emotional state: Exhausted but organized ✅
Grass touching: Imminent ✅