

# PROMPT: versioning strategy v1

## 1. Generic Version Strategy Prompt (AI Project Template)

### Version Strategy for AI-Assisted Projects

#### Core Principle

If a versioning system doesn't have (1) documented increment rules, (2) a tracking location, and (3) a stakeholder who cares, it's not a versioning system—it's a number someone put in a file once.

#### Standard Hierarchy

Level	Format	Increments When	Tracked Where
Product	vX.Y	Major milestone (X), feature set (Y)	Product roadmap
Release	vYYYY.MM.DD	Every deployment	CHANGELOG.md
Document	vX.Y	Major rewrite (X), updates (Y)	Doc meta headers
Component	vX.Y (optional)	Breaking change	File headers

#### Rules

1. **One source of truth per version type** - No duplicate tracking
2. **Versions describe, they don't prescribe** - Track reality, don't enforce ceremony
3. **Stable identifiers ≠ versions** - IDs don't increment (PROJECT-001 stays PROJECT-001)
4. **Dates > semantics for releases** - Calendar versioning for deployment events
5. **Semantic versioning for contracts** - Use X.Y for APIs, schemas, documents

#### Anti-Patterns to Avoid

- ❌ Versions in filenames (use meta headers instead)
- ❌ Time estimates in version/phase names
- ❌ Multiple overlapping version systems without clear hierarchy
- ❌ Versions that nobody updates or checks

#### When to Create a New Version Type

Only if ALL three are true:

1. **Distinct lifecycle** - It changes independently from other versioned things

- 2. **Clear stakeholder** - Someone specific cares about tracking this
- 3. **Increment trigger** - You can define exactly when it bumps

## Adapting This Template

Replace "Product/Release/Document" with your project's specific needs. Common adaptations:

- **Platform > Project > Feature** (for multi-tier systems)
- **Specification > Implementation** (for standards bodies)
- **Schema > Migration** (for data-heavy systems)

## Sugartown CMS Versioning Strategy (Addendum)

### Our Hierarchy

Level	Format	What It Means	Example
Platform	v4.x	Product milestone	v4.2 Stability
Project	PROJ-###	Permanent ID	PROJ-001
PRD	vX.Y	Document version	PRD v1.5
Feature	vX.Y	Deliverable version	Resume Factory v3.0
Release	vYYYY.MM.DD	Deployment event	v2025.01.07
Phase	Phase N	Work breakdown	Phase 1: Foundation
File	vX.Y (optional)	Component maturity	archive-gem.php v6.0

### Key Relationships

- **Platform v4.2** ships across multiple **Releases** (v2025.01.07, v2025.01.14)
- **PROJ-002** (Resume Factory) has **PRD v3.0** documenting **Feature v3.0**
- **Phase 1** work might inform **Platform v4.2** scope but isn't 1:1 mapping
- **File v6.0** increments independently of everything else

### Sugartown-Specific Rules

1. **PRD filenames are stable** - No versions in filenames, only in meta headers
2. **Project IDs never change** - Even if the project pivots or gets renamed
3. **Phases have no time estimates** - "Phase 1: Foundation" not "Phase 1 (Weeks 1-2)"
4. **CHANGELOG.md is release source of truth** - Auto-imported by content\_store.py
5. **Platform versions are strategic** - Not every release bumps the platform version

### Pink Stink Clause

Any reference to "pink stink" in documentation will be immediately deleted. No exceptions.

### Resources

- **Platform roadmap:** sugartown\_design\_system\_PRD.md
- **Release history:** CHANGELOG.md
- **PRD templates:** (see updated PRD prompt)
- **Project registry:** (this doc, Active Projects section)