# **Design Tokens Sprint Tickets & Acceptance Criteria**

**Sprint Duration:** 2 weeks  
**Epic:** Healthcare-Grade Design Tokens Foundation  
**Target:** Single sprint delivery with enterprise automation & semantic versioning

## **🏗️ WEEK 1: Foundation & Architecture**

### **DT-001: Repository Setup & Three-Tier Architecture with Version Control**

**Story Points:** 3  
**Assignee:** Andy  
**Dependencies:** None

#### **User Story**

As a developer, I need a well-structured design tokens repository with proper version control and semantic versioning so I can contribute and consume tokens following industry best practices.

#### **Acceptance Criteria**

* soldev-nyulh-design-tokens repository created with proper structure
* Three-tier folder structure implemented (reference/, system/, component/)
* package.json configured with semantic versioning (starting at v0.1.0)
* Git tags configured for version releases (git tag v0.1.0)
* Conventional commits setup for automated versioning
* .gitignore configured to exclude build/ directory
* Repository permissions set for team access (Alen, Kattia, Huming, Vahan, Mebin)
* Branch protection rules enabled for main branch

#### **Version Control Examples**

# Semantic versioning examples  
v0.1.0 # Initial release with core tokens  
v0.1.1 # Patch: accessibility fixes, contrast improvements  
v0.2.0 # Minor: dark mode tokens, new healthcare categories  
v1.0.0 # Major: Moving Brand migration (breaking changes)

#### **Definition of Done**

* Repository accessible to all team members
* Semantic versioning strategy documented with examples
* Folder structure matches enhanced requirements
* Version control workflow established
* Dependencies install without errors

### **DT-002: Token Extraction & Automated WCAG Validation**

**Story Points:** 5  
**Assignee:** Andy + [@Soul, Joseph](mailto:Joseph.Soul@nyulangone.org) (Design Review)  
**Dependencies:** DT-001

#### **User Story**

As a designer and developer, I need core design tokens extracted from Figma with automated WCAG 2.1 AA compliance testing so all implementations are accessible by default.

#### **Acceptance Criteria**

* Core color tokens extracted from Figma v3/v4 designs
* Typography scale defined with semantic naming (--nyulh-font-size-body-base)
* Spacing system based on 8px grid implemented (--nyulh-space-component-md)
* **Automated a11y testing**: All color combinations validated for 4.5:1 contrast ratio minimum
* **Automated a11y testing**: Pa11y integration for accessibility validation
* Healthcare-specific tokens defined (clinical status, privacy levels)
* Token naming follows semantic conventions with version-safe structure
* JSON structure validates against defined schema
* **Automated a11y testing**: Axe-core integration for continuous compliance

#### **Automated A11y Testing Examples**

# Automated accessibility validation  
npm run test:a11y # Run full accessibility test suite  
npm run test:contrast # Validate color contrast ratios  
npm run test:wcag-aa # WCAG 2.1 AA compliance check  
npm run validate:healthcare # Clinical token appropriateness

#### **Definition of Done**

* Joseph approves extracted token values
* **100% automated a11y validation** passes for all tokens
* All color tokens exceed 4.5:1 contrast ratio
* Healthcare tokens reviewed for clinical appropriateness
* Automated testing pipeline functional

### **DT-003: Enhanced CI/CD Pipeline with Automated A11y & Versioning**

**Story Points:** 5  
**Assignee:** Andy  
**Dependencies:** DT-001

#### **User Story**

As a team, I need automated validation, accessibility testing, and semantic versioning so token changes are tested for compliance and properly versioned before release.

#### **Acceptance Criteria**

* GitHub Actions workflow for PR validation with automated a11y testing
* **Automated semantic versioning** based on conventional commits
* **Automated a11y testing**: WCAG contrast ratio checking in CI
* **Automated a11y testing**: Pa11y and axe-core integration
* Token schema validation in CI pipeline
* Build process generates CSS, SCSS, and JSON outputs
* Automated release creation with proper version tags
* Visual regression testing foundation setup
* Slack notifications for build failures and releases

#### **Semantic Versioning Automation Examples**

# Conventional commit examples for automated versioning  
feat: add dark mode tokens # → v0.2.0 (minor)  
fix: improve contrast ratio for error # → v0.1.1 (patch)   
feat!: rename primary color tokens # → v1.0.0 (major)  
docs: update README with examples # → no version bump

#### **Automated A11y Pipeline**

# CI/CD accessibility validation  
- name: Validate WCAG Compliance  
 run: |  
 npm run test:a11y  
 npm run test:contrast-ratios  
 npm run validate:healthcare-tokens

#### **Definition of Done**

* PR validation includes automated a11y testing
* Semantic versioning works with conventional commits
* All accessibility validations automated
* Release automation functional
* Team receives notifications for failures

## **🚀 WEEK 2: Integration & Distribution**

### **DT-004: Style Dictionary with Enhanced A11y Outputs & Versioning**

**Story Points:** 3  
**Assignee:** Andy  
**Dependencies:** DT-002

#### **User Story**

As a developer, I need optimized build outputs with accessibility metadata and proper versioning so I can integrate tokens efficiently with confidence in compliance.

#### **Acceptance Criteria**

* Style Dictionary configured with custom transforms
* CSS custom properties generated with version comments
* SCSS variables with accessibility metadata included
* JSON output includes WCAG compliance data
* **Automated a11y metadata**: Contrast ratios documented in outputs
* Healthcare-specific mixins and utilities generated
* **Version headers** in all generated files
* Dark mode token variants included (when Joseph provides)
* Build process completes in under 60 seconds

#### **Versioned Output Examples**

/\* Design Tokens v0.1.0 - Generated on 2024-01-15 \*/  
/\* WCAG 2.1 AA Compliant - All contrast ratios validated \*/  
  
:root {  
 --nyulh-color-text-primary: #212121; /\* 15.8:1 contrast ratio \*/  
 --nyulh-color-brand-primary: #5E2E84; /\* 4.61:1 contrast ratio \*/  
 --nyulh-color-clinical-critical: #D32F2F; /\* Emergency status \*/  
}

#### **Definition of Done**

* All platform outputs include version information
* Accessibility metadata embedded in outputs
* Generated files pass automated validation
* Build performance meets requirements

### **DT-005: Drupal Theme Integration & Team Testing**

**Story Points:** 8  
**Assignee:** Andy + Alen + Kattia + Huming + Vahan + Mebin  
**Dependencies:** DT-004

#### **User Story**

As a Drupal development team, I need design tokens integrated into the theme with proper versioning so we can use consistent, accessible design variables across all our components.

#### **Acceptance Criteria**

* Tokens integrated into soldev-contenthub-drupal-theme-movingbrand
* **Version synchronization**: Theme version updated to coordinate with token version
* CSS custom properties available in Twig templates
* SCSS variables available in component stylesheets
* Healthcare tokens tested in realistic medical UI contexts
* Integration tested by full dev team (Alen, Kattia, Huming, Vahan, Mebin)
* **Automated a11y testing**: Drupal theme passes accessibility validation
* No breaking changes to current development workflow
* Example components demonstrate token usage with versions

#### **Version Coordination Example**

// Theme package.json coordination  
{  
 "name": "soldev-contenthub-drupal-theme-movingbrand",  
 "version": "2.1.0", // Theme version  
 "dependencies": {  
 "@nyulangone/design-tokens": "^0.1.0" // Token dependency  
 }  
}

#### **Team Testing Assignments**

* **Alen**: SCSS integration and build process testing
* **Kattia**: Twig template integration and component usage
* **Huming**: Healthcare token validation and clinical context testing
* **Vahan**: Cross-browser compatibility and accessibility testing
* **Mebin**: Performance impact and build optimization testing

#### **Definition of Done**

* All team members confirm integration works in their contexts
* Theme builds successfully with versioned tokens
* **Automated a11y tests pass** for integrated components
* Version coordination documented and functional
* No regressions in existing functionality

### **DT-006: Semantic Versioning Strategy & Migration Tools**

**Story Points:** 4  
**Assignee:** Andy  
**Dependencies:** DT-003, DT-004

#### **User Story**

As a team member, I need comprehensive versioning strategy with automated migration tools so I understand how to safely update design tokens and coordinate with theme updates.

#### **Acceptance Criteria**

* **Complete semantic versioning documentation** with healthcare-specific examples
* **Automated versioning** based on conventional commit messages
* **Migration tools framework** created for future breaking changes
* **Version compatibility matrix** documented for token/theme coordination
* **Cross-repository automation** configured for dependency updates
* **Deprecation timeline** defined (18-month healthcare cycle)
* **Release channels** configured (alpha, beta, stable)

#### **Semantic Versioning Examples with Healthcare Context**

# Version bump examples  
v0.1.0 → v0.1.1 # PATCH: Fix contrast ratio for clinical warning color  
v0.1.1 → v0.2.0 # MINOR: Add new privacy level tokens for HIPAA contexts  
v0.2.0 → v1.0.0 # MAJOR: Moving Brand migration (breaking token renames)  
  
# Release channel strategy  
v1.0.0-alpha.1 # Alpha: Moving Brand preview for early testing  
v1.0.0-beta.1 # Beta: Moving Brand feature-complete testing  
v1.0.0-rc.1 # RC: Moving Brand production readiness validation  
v1.0.0 # Stable: Moving Brand production release

#### **Migration Tools Framework**

# Automated migration commands (future v1.0.0)  
npm run tokens:migrate:preview # Show what will change  
npm run tokens:migrate:apply # Apply automated changes  
npm run tokens:migrate:validate # Verify migration success  
npm run tokens:migrate:rollback # Undo if needed

#### **Definition of Done**

* Semantic versioning fully automated and documented
* Migration tools framework tested with sample scenarios
* Version compatibility matrix comprehensive
* Release process validated end-to-end

### **DT-007: Documentation & Team Training with Versioning Focus**

**Story Points:** 3  
**Assignee:** Andy + Teresa (Training Coordination)  
**Dependencies:** DT-005, DT-006

#### **User Story**

As a team member, I need comprehensive documentation and training covering versioning strategy so I can effectively use and maintain design tokens.

#### **Acceptance Criteria**

* Enhanced README with versioning examples and healthcare context
  + Include PHPStorm and VSCode tools for CSS/SCSS design token completion
    - <https://www.css-variables-assistant.dev/>
    - <https://marketplace.visualstudio.com/items?itemName=vunguyentuan.vscode-css-variables>
* **Semantic versioning guide** with practical examples
* **Automated a11y testing documentation** with usage examples
* Healthcare implementation examples with version coordination
* RACI matrix for governance and version approval authority
* Contribution guidelines including versioning requirements
* **60-minute team training session** covering usage + versioning + a11y
* Quick reference guide for developers with version commands

#### **Training Session Agenda (60 minutes)**

1. **Design tokens overview** (10 min) - What and why
2. **Three-tier architecture** (10 min) - Reference/System/Component
3. **Semantic versioning strategy** (15 min) - Examples and automation
4. **Accessibility testing** (10 min) - Automated validation
5. **Healthcare tokens** (10 min) - Clinical context and privacy
6. **Hands-on usage** (15 min) - Live integration examples

#### **Definition of Done**

* All documentation comprehensive and version-aware
* Team training includes versioning and a11y focus
* Quick reference accessible with version commands
* Governance roles include version authority
* Contribution process includes versioning requirements

## **🏥 SUPPORTING TICKETS**

### **DT-008: Healthcare Token Validation with Huming**

**Story Points:** 2  
**Assignee:** Andy + [@Hopper, Cody](mailto:Cody.Hopper@nyulangone.org) (Accessibility) + Huming (Clinical Validation)  
**Dependencies:** DT-002

#### **User Story**

As Huming (acting as clinical stakeholder), I need design tokens that appropriately represent healthcare contexts so medical interfaces communicate status clearly and safely.

#### **Acceptance Criteria**

* Clinical status tokens validated by Huming for medical appropriateness
* Privacy level indicators comply with HIPAA considerations
* Color choices validated for medical environments by Huming
* **Automated a11y testing** validated with assistive technologies
* Documentation includes healthcare-specific usage guidelines
* Version strategy accounts for clinical context stability requirements

#### **Definition of Done**

* Huming's clinical approval documented
* Cody confirms accessibility compliance
* Healthcare usage guide complete with versioning considerations

### **DT-009: Automated A11y Testing Integration**

**Story Points:** 3  
**Assignee:** Andy + Cody  
**Dependencies:** DT-003

#### **User Story**

As a team, I need comprehensive automated accessibility testing so we maintain WCAG 2.1 AA compliance automatically across all token changes.

#### **Acceptance Criteria**

* **Pa11y integration** for automated accessibility auditing
* **Axe-core integration** for comprehensive a11y validation
* **Contrast ratio automation** for all color token combinations
* **WCAG 2.1 AA compliance reporting** in CI/CD pipeline
* **Healthcare accessibility standards** validation
* **Automated regression prevention** for accessibility issues
* **Version-aware testing** ensuring compliance across token updates

#### **Automated A11y Testing Suite**

# Comprehensive accessibility testing  
npm run test:a11y:full # Complete accessibility audit  
npm run test:a11y:contrast # Color contrast validation  
npm run test:a11y:healthcare # Healthcare-specific checks  
npm run test:a11y:regression # Prevent accessibility regressions  
npm run test:a11y:ci # CI-optimized testing

#### **Definition of Done**

* All automated a11y tests integrated and passing
* CI/CD pipeline includes comprehensive accessibility validation
* Regression prevention functional
* Healthcare accessibility standards validated

## **📊 SPRINT SUMMARY**

### **Total Story Points: 36**

### **Critical Path: DT-001 → DT-002 → DT-004 → DT-005 → DT-007**

### **Enhanced Focus: Semantic Versioning + Automated A11y Testing**

### **Success Metrics**

* **100% automated WCAG 2.1 AA compliance** for all tokens
* **Semantic versioning fully operational** with examples
* **Zero breaking changes** to current workflow
* **Healthcare context validated** by Huming
* **Automated a11y pipeline** prevents regressions
* **Version coordination** between repositories functional
* **Migration tools framework** ready for Moving Brand

### **Team Dependencies & Coordination**

* **Joseph**: 6-8 hours for design validation and healthcare context review
* **Cody**: 6-8 hours for accessibility validation and automated testing setup
* **Andy**: Full sprint technical lead (repository, CI/CD, integration, versioning)
* **Alen**: 3-4 hours for SCSS integration and build process testing
* **Kattia**: 3-4 hours for Twig template integration testing
* **Huming**: 4-5 hours for healthcare token validation and clinical context review
* **Vahan**: 2-3 hours for cross-browser compatibility and accessibility testing
* **Mebin**: 2-3 hours for performance impact and build optimization testing
* **Teresa**: Sprint coordination, training scheduling, RACI facilitation

### **Enhanced Risk Mitigation**

* **Joseph Availability**: Core tokens extracted first, healthcare refinements in parallel
* **CI/CD Complexity**: Automated a11y testing prioritized, advanced features iterative
* **Integration Issues**: Manual fallback always available, multiple team members testing
* **Version Complexity**: Clear examples and documentation, automated wherever possible
* **Timeline Pressure**: MVP approach with comprehensive post-sprint roadmap

### **Version Control & Semantic Versioning Highlights**

* **v0.1.0**: Sprint completion with core tokens and automation
* **Conventional commits**: Automated version bumps based on commit messages
* **Release channels**: Alpha/beta/stable pipeline for Moving Brand transition
* **Migration tools**: Framework ready for v1.0.0 breaking changes
* **Cross-repo coordination**: Theme versions synchronized with token releases

### **Automated A11y Testing Highlights**

* **Pa11y + Axe-core**: Comprehensive accessibility validation
* **Contrast ratio automation**: All color combinations validated
* **CI/CD integration**: Accessibility regression prevention
* **Healthcare standards**: Clinical context accessibility validation
* **Version-aware testing**: Compliance maintained across updates

### **Post-Sprint Handoff**

* **v0.1.0 released** with full semantic versioning and automated a11y
* **Team trained** on versioning strategy and accessibility testing
* **Governance established** with version authority and a11y standards
* **Automation functional** for ongoing development
* **Roadmap defined** for Moving Brand migration using established patterns