

2025-11-01

# E020: Version Control and Git Workflow

DIP-SMC-PSO Educational Series

January 25, 2026

## Overview

This episode covers version control and git workflow from the DIP-SMC-PSO project.

**Part:** Part4 Professional

**Duration:** 15-20 minutes

**Source:** Comprehensive Presentation Materials

## section0 Intentional Architectural Patterns

**\*\*DO NOT "FIX" These Patterns – They Are Intentional:\*\***

- **Compatibility Layers**
- ‘src/optimizer/‘ → ‘src/optimization/‘ - Backward compatibility for old imports - *Reason:* Smooth migration, no breaking changes
- **Re-export Chains**
- ‘simulation\_context.py‘ in 3 locations - Import path flexibility - *Reason:* User convenience, multiple valid import styles
- **Model Variants**
- 8 dynamics files (simplified, full, low-rank, etc.) - Different accuracy/performance tradeoffs
- *Reason:* Speed vs. accuracy based on use case
- **Framework Files in ‘src/\*\*‘**
- ‘src/interfaces/hil/test\_automation.py‘ is PRODUCTION code - Automation framework, not pytest tests - *Reason:* Exported in ‘\_\_init\_\_.py‘, imported by production code

## section0 Directory Placement Rules

**\*\*File Classification Decision Tree:\*\***

[Visual diagram - see PDF]

## section0 Quality Gates (ENFORCE STRICTLY)

**\*\*Production Readiness Criteria:\*\***

	<b>**Gate**</b>	<b>**Threshold**</b>	<b>**Current**</b>
Critical issues	0		[OK]
High-priority issues	3		[OK]
Test pass rate [OK] (14)	100 Root visible items		19
Malformed file names	0		[OK]
Code coverage (overall) 70/100	85 Coverage (critical) (23.9)	95 Production score	

**\*\*Research-Ready:\*\*** [OK] Single/multi-threaded operation validated

\*NOT Production-Ready.\* Quality gates 1/8 passing (score: 23.9/100)

*See:* ‘.ai\_workspace/guides/phase4\_status.md‘

## section0 File Naming Conventions

**\*\*NEVER Create These Patterns:\*\***

- **Braces/spaces:** ‘{dir}‘, ‘my folder/‘ - **Windows device names:** ‘nul‘, ‘con‘, ‘prn‘, ‘aux‘ - **Unicode paths on Windows:** Use ASCII only - **Trailing dots/spaces:** ‘file.‘, ‘dir.‘

**\*\*Recommended Patterns:\*\***

- **Python modules:** ‘snake\_case.py‘ - **Classes:** ‘PascalCase‘ - **Functions/variables:** ‘snake\_case‘ - **Constants:** ‘UPPER\_SNAKE\_CASE‘ - **Directories:** ‘lowercase\_underscores/‘ - **Scripts:** ‘verb\_noun.sh‘ (e.g., ‘run\_tests.sh‘)

## Resources

- **Repository:** <https://github.com/theSadeQ/dip-smc-pso.git>
- **Documentation:** See docs/ directory
- **Getting Started:** docs/guides/getting-started.md

*Educational podcast episode generated from comprehensive presentation materials*