

Day 2: Intermediate Application with Archive Integration

GitHub Spec Kit Intensive Training - Intermediate Day

Duration: 6-8 hours

Objective: Apply SDD to real HX-Infrastructure scenarios, master archive integration, develop intermediate proficiency

Success Criteria: Complete HX-Infrastructure project using Spec Kit, demonstrate 70-80% autonomous proficiency



Morning Session (3-4 hours)

Hour 1: Day 1 Review & HX-Infrastructure Analysis

1.1 Foundation Skills Validation (30 minutes)

Quick Proficiency Check:

```
# Verify yesterday's work is accessible
cd /home/ubuntu/github_spec_training
ls -la day1_learning_project/

# Test Spec Kit commands still work
cd day1_learning_project
# In your AI agent, test these commands:
# /specify --help
# /plan --help
# /tasks --help
# /implement --help
```

Foundation Review Checklist:

- [] Can initialize new Spec Kit projects independently
- [] Understands all four SDD phases conceptually
- [] Successfully used AI agent integration yesterday
- [] Completed basic specification and implementation
- [] Environment remains stable and functional

1.2 HX-Infrastructure Deep Dive (30 minutes)

Archive Analysis Exercise:

```
cd /home/ubuntu/github_spec_training/HX-Infrastructure-Knowledge-Base

# Analyze repository structure
tree -L 3 . || find . -type d | head -20

# Look for project patterns
grep -r -i "project\|task\|workflow" . --include="*.md" | head -10

# Identify common themes and categories
find . -name "*.md" -exec basename {} \; | sort | uniq -c | sort -nr | head -10
```

Key Analysis Questions:

1. What types of projects are documented in the archive?
2. What common workflows and processes are evident?
3. What pain points or challenges are mentioned repeatedly?
4. How could Spec-Driven Development address these challenges?
5. What integration opportunities exist with current tools?

Hour 2: Real-World Project Specification

2.1 Project Selection and Scoping (45 minutes)

Choose a Real HX-Infrastructure Challenge:

Based on your archive analysis, select one of these project types:

1. Knowledge Base Enhancement System

- Automated documentation generation
- Cross-reference linking
- Search and discovery improvements

1. Infrastructure Monitoring Dashboard

- Real-time system status
- Alert management
- Performance analytics

2. Project Workflow Automation

- Task assignment and tracking
- Progress reporting
- Integration with existing tools

3. Documentation Quality Assurance

- Automated content validation
- Style and format consistency
- Link checking and maintenance

Project Scoping Framework:

- **Scope:** What's included and excluded
- **Stakeholders:** Who will use and benefit from this
- **Success Metrics:** How you'll measure success
- **Constraints:** Time, resources, technical limitations
- **Assumptions:** What you're assuming to be true

2.2 Advanced Specification Creation (15 minutes)

Create a new Spec Kit project for your chosen real-world challenge:

```
cd /home/ubuntu/github_spec_training
uvx --from git+https://github.com/github/spec-kit.git specify init hx_infrastructure_p
roject --ai copilot
cd hx_infrastructure_project
```

Use the `/specify` command with enhanced context from your archive analysis.

Hour 3: Advanced Planning Techniques

3.1 Context-Rich Planning (45 minutes)

Enhanced Planning Process:

1. Stakeholder Analysis:

- Primary users (HX-Infrastructure team members)
- Secondary users (external collaborators)
- System administrators and maintainers
- Future developers and contributors

2. Technical Constraints Integration:

- Existing HX-Infrastructure technology stack
- Security and compliance requirements
- Performance and scalability needs
- Integration with current tools and workflows

3. Risk Assessment and Mitigation:

- Technical risks and mitigation strategies
- Resource availability and constraints
- Timeline pressures and dependencies
- Change management and adoption challenges

Use the `/plan` command with comprehensive context:

- Include findings from your archive analysis
- Reference specific HX-Infrastructure requirements
- Consider integration with existing systems
- Plan for scalability and maintenance

3.2 Plan Validation and Stakeholder Review (15 minutes)

Plan Quality Assessment:

- [] Addresses real HX-Infrastructure needs identified in archive
 - [] Technology choices align with existing infrastructure
 - [] Security and compliance requirements considered
 - [] Integration points clearly defined
 - [] Scalability and maintenance planned
 - [] Risk mitigation strategies included
-

Afternoon Session (3-4 hours)

Hour 4: Advanced Task Management

4.1 Complex Task Breakdown (45 minutes)

Advanced Task Management Techniques:

1. Dependency Mapping:

- Identify critical path tasks
- Map inter-task dependencies
- Plan for parallel execution
- Account for external dependencies

2. Risk-Based Prioritization:

- High-risk tasks tackled early
- Proof-of-concept for uncertain areas
- Integration points validated first
- User feedback loops built in

3. Resource Allocation:

- Tasks matched to skill levels
- Time estimates with buffers
- Review and validation points
- Knowledge transfer requirements

Use `/tasks` command with advanced context:

- Break down into 2-4 hour tasks
- Include validation and testing tasks
- Plan for documentation and knowledge transfer
- Consider deployment and maintenance tasks

4.2 Task Validation and Refinement (15 minutes)

Task Quality Checklist:

- [] Each task has clear acceptance criteria
- [] Dependencies are explicitly mapped
- [] Risk levels are assessed and mitigated
- [] Time estimates include buffers
- [] Validation methods are defined
- [] Knowledge transfer is planned

Hour 5: Implementation with Archive Integration

5.1 Implementation Strategy (30 minutes)

Advanced Implementation Approach:

1. Archive-Informed Development:

- Use existing HX-Infrastructure patterns
- Leverage documented best practices
- Avoid known pitfalls and issues
- Build on successful approaches

2. Iterative Validation:

- Implement in small increments

- Validate against archive examples
- Get feedback from stakeholders
- Refine based on real-world usage

3. Knowledge Distillation:

- Extract lessons from archive content
- Document new insights and patterns
- Create reusable components
- Build institutional knowledge

5.2 Hands-On Implementation (30 minutes)

Implementation Sprint:

Implement the first 3-4 critical tasks from your task list:

Focus Areas:

1. Core Architecture Setup

- Basic project structure
- Integration with HX-Infrastructure patterns
- Configuration and environment setup

1. Key Functionality Implementation

- Primary user workflows
- Data models and persistence
- Basic user interface

2. Integration Points

- Connections to existing systems
- Data import/export capabilities
- API endpoints or interfaces

Implementation Validation:

- Code follows HX-Infrastructure standards
- Integration points work as designed
- Basic functionality meets specifications
- Documentation is clear and complete

Hour 6: Quality Assurance and Testing

6.1 Specification-Driven Testing (45 minutes)

Testing Strategy:

1. Specification Validation:

- Test against original specifications
- Verify user stories are satisfied
- Check acceptance criteria compliance
- Validate success metrics

2. Integration Testing:

- Test connections to existing systems
- Verify data flow and transformations
- Check error handling and recovery
- Validate security and permissions

3. User Experience Testing:

- Test with realistic HX-Infrastructure scenarios
- Verify workflows match user expectations
- Check performance under realistic loads
- Validate accessibility and usability

Testing Implementation:

- Create test cases based on specifications
- Use archive content for realistic test data
- Implement automated tests where possible
- Document test results and findings

6.2 Feedback Integration (15 minutes)**Feedback Loop Implementation:**

- Document testing results
- Identify areas for improvement
- Plan refinements and iterations
- Update specifications based on learnings

**Evening Session (1-2 hours)****Hour 7: Knowledge Distillation and Documentation****7.1 Lessons Learned Extraction (45 minutes)****Knowledge Distillation Process:****1. Archive Insights:**

- What patterns worked well in past projects?
- What challenges were commonly encountered?
- How can Spec-Driven Development address these?
- What new patterns are emerging?

2. Implementation Learnings:

- What worked well in today's implementation?
- What challenges did you encounter?
- How did AI assistance help or hinder?
- What would you do differently next time?

3. Process Improvements:

- How can the SDD process be optimized?
- What tools or techniques would be helpful?
- How can knowledge transfer be improved?
- What documentation standards should be adopted?

Create Knowledge Base Entry:

```
# Create lessons learned document
touch /home/ubuntu/github_spec_training/day2_knowledge_distillation.md
```

Document your insights in a structured format that can be used for teaching others.

7.2 Reusable Component Creation (15 minutes)

Component Library Development:

- Extract reusable patterns from today's work
- Create templates for common HX-Infrastructure scenarios
- Document best practices and guidelines
- Build foundation for teaching materials

Hour 8: Day 2 Validation and Day 3 Preparation

8.1 Intermediate Proficiency Assessment (30 minutes)

Intermediate Skills Checklist:

- ☐ Can analyze existing projects for SDD opportunities
- ☐ Creates comprehensive specifications for real-world projects
- ☐ Integrates archive content and lessons learned effectively
- ☐ Develops complex technical plans with AI assistance
- ☐ Breaks down large projects into manageable tasks
- ☐ Implements solutions that integrate with existing systems
- ☐ Validates work against specifications consistently
- ☐ Documents learnings for knowledge transfer

Proficiency Self-Assessment:

- **Intermediate (61-80%):** Can work independently on most tasks
- **Advanced (81-90%):** Can handle complex scenarios with minimal guidance
- **Expert (91-100%):** Can teach others and optimize processes

Target for Day 2: 70-80% proficiency in intermediate skills

8.2 Day 3 Preparation (15 minutes)

Tomorrow's Focus: Advanced Techniques and Optimization

Preparation Tasks:

- ☐ Review today's implementation for optimization opportunities
- ☐ Identify advanced Spec Kit features to explore
- ☐ Prepare complex scenarios for advanced practice
- ☐ Set up environment for performance testing and optimization

8.3 Progress Documentation (15 minutes)

Day 2 Progress Summary:

- Real-world project selected and scoped
 - Comprehensive specification created with archive integration
 - Advanced planning techniques applied
 - Complex task breakdown completed
 - Implementation with HX-Infrastructure integration
 - Quality assurance and testing performed
 - Knowledge distillation and documentation completed
-

Day 2 Success Validation

Mandatory Completion Criteria:

- [] Real HX-Infrastructure project selected and analyzed
- [] Comprehensive specification created with archive insights
- [] Advanced technical plan developed
- [] Complex task breakdown completed
- [] Core functionality implemented and tested
- [] Integration with existing systems demonstrated
- [] Knowledge distillation documented
- [] Lessons learned captured for teaching

Proficiency Indicators:

- Can apply SDD to real-world scenarios independently
- Integrates archive content effectively into new projects
- Uses AI assistance efficiently for complex tasks
- Validates work against specifications consistently
- Documents learnings for knowledge transfer
- Ready for advanced optimization techniques

If You're Behind Schedule:

- Focus on completing the specification and planning phases
- Implement core functionality even if not all features
- Ensure integration concepts are understood
- Document key learnings even if implementation is incomplete
- Plan catch-up time for Day 3 morning session



Additional Resources for Day 2

Advanced Reading:

- GitHub Spec Kit advanced features documentation
- HX-Infrastructure architecture patterns
- Integration best practices and patterns

Practical Resources:

- Archive analysis tools and techniques
- Testing frameworks and methodologies
- Documentation standards and templates

Community Resources:

- Spec-driven development case studies
 - Integration pattern libraries
 - Best practices from similar projects
-

End of Day 2

Next: Day 3 - Advanced Techniques and Optimization

Estimated Completion Time: 6-8 hours

Success Rate Target: 70-80% proficiency in intermediate skills

You're making excellent progress! Day 2 builds real-world application skills that form the foundation for advanced techniques. Focus on quality over quantity, and ensure you understand the integration concepts before moving forward.