Day 1: Foundation & Setup Mastery

GitHub Spec Kit Intensive Training - Foundation Day

Duration: 6-8 hours
Objective: Master environment setup, understand SDD fundamentals, complete first spec-driven project
Success Criteria: 100% environment validation, successful project initialization, basic workflow proficiency

🎯 **HX-Infrastructure Integration:** Project setup, repository analysis, and initial knowledge base structure creation

---

🌅 Morning Session (3-4 hours)

Hour 1: Environment Validation & Setup

1.1 Complete Environment Validation (30 minutes)

```
# Run comprehensive validation
cd /home/ubuntu/github_spec_training
./validate_environment.sh

# If any failures, address immediately
./validate_environment.sh --install-help
```

Validation Checklist:
- [ ] Python 3.11+ installed and accessible
- [ ] Git configured with your credentials
- [ ] UV package manager working
- [ ] WSL2 properly configured (Windows 11)
- [ ] AI coding agent accessible (GitHub Copilot/Claude Code)
- [ ] Network connectivity to GitHub and PyPI
- [ ] HX-Infrastructure-Knowledge-Base repository cloned

1.2 GitHub Spec Kit Installation & Verification (30 minutes)

```
# Test Spec Kit installation
uvx --from git+https://github.com/github/spec-kit.git specify init test_foundation_pro
ject

# Verify installation success
cd test_foundation_project
ls -la
cat README.md
```

Expected Outputs:
- Project directory created with proper structure
- Configuration files present (.specify/, prompts/, etc.)

- AI agent integration working
- Slash commands accessible in your AI agent

Hour 2: Spec-Driven Development Fundamentals

2.1 Understanding the SDD Philosophy (45 minutes)

Core Concepts to Master:
1. **Intent-First Development:** Specifications capture “what” and “why” before “how”
2. **AI-Assisted Implementation:** Leverage AI agents for code generation from clear specs
3. **Iterative Refinement:** Specifications evolve through feedback loops
4. **Quality Through Clarity:** Better specs = better code = fewer bugs

🎯 **HX-Infrastructure Practical Exercise:**

```
# Clone and analyze the HX-Infrastructure Knowledge Base
cd /home/ubuntu/github_spec_training
git clone https://github.com/hanax-ai/HX-Infrastructure-Knowledge-Base.git
cd HX-Infrastructure-Knowledge-Base

# Create your first specification for knowledge base enhancement
uvx --from git+https://github.com/github/spec-kit.git specify init hx_kb_enhancement -
-ai copilot
```

In your AI agent, use the `/specify` command to create a specification for:
Project: “HX-Infrastructure Knowledge Base Content Integration”

Specification Requirements:
- Analyze current knowledge base structure and content gaps
- Create systematic approach for integrating archived project learnings
- Establish documentation standards and templates
- Design validation workflows for content quality
- Plan progressive content population across training days

2.2 The Four Phases Deep Dive (30 minutes)

Phase 1: Specify
- Define clear, actionable requirements
- Establish success criteria and constraints
- Document assumptions and dependencies

Phase 2: Plan
- Break down specifications into implementable tasks
- Sequence work for optimal flow
- Identify potential risks and mitigation strategies

Phase 3: Implement
- Execute planned tasks with AI assistance
- Maintain quality through continuous validation
- Document decisions and learnings

Phase 4: Validate
- Test implementations against specifications
- Gather feedback and iterate
- Prepare for next development cycle

🎯 **HX-KB Application:**
Apply these phases to analyze the current HX-Infrastructure Knowledge Base:
1. **Specify:** Document what content needs to be integrated
2. **Plan:** Create integration roadmap for training week
3. **Implement:** Begin with directory structure and templates
4. **Validate:** Ensure structure aligns with integration plan

Hour 3: First Real Project Implementation

3.1 HX-Infrastructure Knowledge Base Analysis (45 minutes)

Current State Assessment:

```
cd HX-Infrastructure-Knowledge-Base

# Analyze current structure
find . -type f -name "*.md" | head -20
cat README.md | head -50

# Review existing workflow
cat .github/workflows/connectivity-check.yml
```

Analysis Tasks:
1. **Repository Structure Review:**
- Document current directory layout
- Identify placeholder sections that need content
- Note existing documentation patterns

1. **Content Gap Analysis:**
   - List sections marked as placeholders
   - Identify missing documentation categories
   - Assess integration opportunities

2. **Workflow Assessment:**
   - Review existing GitHub Actions workflow
   - Identify enhancement opportunities
   - Plan additional validation workflows

🎯 **Deliverable:** Create `docs/analysis/day1-assessment.md` documenting findings

3.2 Initial Knowledge Base Enhancement (30 minutes)

Create Foundation Structure:

```
# Create initial directory structure for integration
mkdir -p docs/analysis
mkdir -p docs/integration
mkdir -p exercises/hx-kb
mkdir -p metrics/training

# Create first ADR for integration approach
mkdir -p docs/adrs
```

First ADR Creation:
Create `docs/adrs/ADR-0001-training-integration.md` with:
- **Status:** Proposed

- **Context:** Integration of HX-KB into training program
- **Decision:** Use progressive content development approach
- **Consequences:** Enhanced practical learning, real project outcomes

3.3 Specification Refinement (15 minutes)

Refine Your Specification:
Based on analysis, update your specification to include:
- Specific content integration priorities
- Training day milestone mapping
- Quality validation criteria
- Success measurement approaches

🌆 Afternoon Session (3-4 hours)

Hour 4: Advanced Specification Techniques

4.1 Multi-Stakeholder Specifications (45 minutes)

Understanding Stakeholder Perspectives:
- **End Users:** Team members who will use the knowledge base
- **Contributors:** Developers who will add content
- **Maintainers:** Those responsible for keeping content current
- **Trainers:** Instructors using the knowledge base for education

🎯 **HX-KB Stakeholder Analysis:**
Create specifications that address each stakeholder’s needs:

1. **End User Specification:**
   - Quick access to relevant information
   - Clear navigation and search capabilities
   - Practical examples and templates

2. **Contributor Specification:**
   - Simple contribution workflow
   - Clear documentation standards
   - Automated validation and feedback

3. **Maintainer Specification:**
   - Content freshness monitoring
   - Quality assurance processes
   - Update notification systems

4. **Trainer Specification:**
   - Progressive learning materials
   - Practical exercises and examples
   - Assessment and validation tools

4.2 Specification Validation Techniques (30 minutes)

Validation Methods:
1. **Stakeholder Review:** Get feedback from each stakeholder group
2. **Prototype Testing:** Build minimal viable implementations

3. **Scenario Walkthrough:** Test specifications against real use cases
4. **Constraint Validation:** Ensure specifications are achievable

🎯 **HX-KB Validation Exercise:**
Validate your knowledge base specifications by:
1. Walking through a typical user journey
2. Testing the contribution workflow
3. Verifying maintenance procedures
4. Confirming training integration points

Hour 5: Implementation Planning

5.1 Task Breakdown and Sequencing (45 minutes)

Breaking Down Complex Specifications:
- Identify atomic, implementable tasks
- Establish dependencies between tasks
- Sequence for optimal development flow
- Estimate effort and complexity

🎯 **HX-KB Implementation Plan:**
Create detailed task breakdown for knowledge base integration:

Phase 1 Tasks (Day 1-2):
- [ ] Complete directory structure creation
- [ ] Develop documentation templates
- [ ] Create initial ADRs
- [ ] Set up basic validation workflows

Phase 2 Tasks (Day 2-3):
- [ ] Populate sprint summaries
- [ ] Create operational runbooks
- [ ] Integrate architecture documentation
- [ ] Enhance CI/CD workflows

Phase 3 Tasks (Day 3-4):
- [ ] Add automation guides
- [ ] Create troubleshooting documentation
- [ ] Integrate security best practices
- [ ] Develop metrics and tracking

Phase 4 Tasks (Day 4-5):
- [ ] Complete content integration
- [ ] Validate all documentation
- [ ] Create training materials
- [ ] Establish maintenance procedures

5.2 Risk Assessment and Mitigation (30 minutes)

Common Implementation Risks:
- Scope creep and over-engineering
- Inconsistent documentation standards
- Integration complexity
- Time constraints

🎯 **HX-KB Risk Mitigation:**
Identify and plan mitigation for:
1. **Content Quality Risk:** Establish review processes
2. **Integration Complexity:** Use incremental approach
3. **Time Management:** Prioritize high-value content
4. **Stakeholder Alignment:** Regular check-ins and feedback

Hour 6: Quality Assurance and Documentation

6.1 Specification Quality Checklist (30 minutes)

Quality Criteria:
- [ ] Clear and unambiguous language
- [ ] Measurable success criteria
- [ ] Realistic constraints and assumptions
- [ ] Comprehensive stakeholder coverage
- [ ] Implementable task breakdown
- [ ] Risk assessment and mitigation

🎯 **HX-KB Quality Review:**
Review your specifications against quality criteria and refine as needed.

6.2 Documentation and Handoff (45 minutes)

Documentation Requirements:
1. **Specification Document:** Complete, validated specification
2. **Implementation Plan:** Detailed task breakdown and timeline
3. **Risk Register:** Identified risks and mitigation strategies
4. **Stakeholder Map:** Key contacts and responsibilities

🎯 **Day 1 Deliverables:**
Create and commit the following to your HX-KB repository:
- `docs/analysis/day1-assessment.md` - Current state analysis
- `docs/adrs/ADR-0001-training-integration.md` - Integration decision record
- `docs/integration/implementation-plan.md` - Detailed implementation roadmap
- `docs/integration/stakeholder-analysis.md` - Stakeholder needs and requirements

6.3 Day 1 Validation and Wrap-up (15 minutes)

Validation Checklist:
- [ ] Environment fully validated and working
- [ ] Spec Kit installation verified
- [ ] HX-KB repository analyzed and documented
- [ ] Initial specifications created and validated
- [ ] Implementation plan developed
- [ ] Day 1 deliverables committed to repository

Success Metrics:
- Specification quality score: Target 85%+
- Task breakdown completeness: 100%
- Stakeholder coverage: All groups addressed
- Documentation standards: Consistent formatting and structure

📚 Resources and References

Essential Reading

- GitHub Spec Kit Documentation (https://github.com/github/spec-kit)
- HX-Infrastructure Knowledge Base (https://github.com/hanax-ai/HX-Infrastructure-Knowledge-Base)
- Specification-Driven Development Guide (docs/sdd-guide.md)

Templates and Examples

- ADR Template (templates/adr-template.md)
- Specification Template (templates/specification-template.md)
- Implementation Plan Template (templates/implementation-plan-template.md)

Tools and Utilities

- UV Package Manager
- GitHub Spec Kit CLI
- AI Coding Assistants (Copilot/Claude)
- Validation Scripts

🎯 Day 1 Success Criteria

Technical Proficiency:
- [ ] 100% environment validation passed
- [ ] Spec Kit commands working correctly
- [ ] HX-KB repository successfully analyzed
- [ ] Initial specifications created and validated

Project Outcomes:
- [ ] Current state assessment completed
- [ ] Integration approach documented
- [ ] Implementation plan created
- [ ] Foundation structure established

Knowledge Retention:
- [ ] SDD philosophy understood and articulated
- [ ] Four-phase process demonstrated
- [ ] Stakeholder analysis completed
- [ ] Quality validation performed

Next Day Preparation:
- [ ] Day 2 objectives reviewed
- [ ] Required materials prepared
- [ ] Implementation environment ready
- [ ] Team coordination confirmed

Continue to Day 2: Intermediate Application (day2_intermediate.md) →