next8n

SOP: DEVELOPER

Workflow Automation Delivery Framework

ENTERPRISE EDITION

Version: 2.0

Date: December 28, 2025

Author: Mirza Iqbal

Contact: mirza.iqbal@next8n.com

Table of Contents

Table of Contents

SOP: Developer

Standard Operating Procedure for Building, Testing & Documentation

Role Overview

Daily Workflow

Morning Routine (15 min)

Core Activities

Development Standards

Workflow Structure

Naming Conventions

Error Handling Standards

Code in Code Nodes

Building Workflows

Step-by-Step Process

AI Development Standards

Testing Requirements

Self-Testing Checklist

Test Log Template

Documentation Requirements

In-Workflow Documentation

Technical Documentation

Communication

Daily Standup Format

Asking for Help

Reporting Issues

Handoff to QA

Pre-QA Checklist

Best Practices

Development Best Practices

Efficiency Tips

Common Patterns

Error Handling Pattern

Logging Pattern

AI Processing Pattern

Metrics

Personal Tracking

SOP: Developer

Standard Operating Procedure for Building, Testing & Documentation

Role Overview

Title: n8n Developer / Automation Developer
Reports To: Technical Lead
Receives From: Technical Lead (via Project Manager)
Hands Off To: Technical Lead (for QA) Project Manager (for handover)

Primary Objective:
Build high-quality, well-tested, and documented automation workflows.

Daily Workflow

Morning Routine (15 min)

```
Review assigned tasks
Check for Technical Lead feedback
Review any client notes
Plan the day's work
Flag any blockers early
```

Core Activities

```
1. DEVELOPMENT (4-6 hours)
    Build workflows
    Configure integrations
    Implement logic
    Handle errors
    Test as you go

2. TESTING (1-2 hours)
    Internal testing
    Edge cases
    Error scenarios
    Log results

3. DOCUMENTATION (30 min - 1 hour)
    Add sticky notes
    Update docs
    Record videos (if needed)

4. COMMUNICATION (30 min)
    Daily sync with Tech Lead
    Report progress
    Ask questions
    Update task status
```

Development Standards

Workflow Structure

```
WORKFLOW ORGANIZATION:

LEFT TO RIGHT FLOW:
Triggers  Processing  AI  Actions  Outputs

VERTICAL GROUPING:
Group related nodes vertically

STICKY NOTES:
Add to explain:
- What this section does
- Why certain choices made
- Edge case handling
- Important notes
```

Naming Conventions

```
NODE NAMING FORMAT:
[Action] [Target]

EXAMPLES:
 "Get Customer from HubSpot"
 "Send Welcome Email"
 "Update CRM Record"
 "Parse API Response"
 "Check if Exists"
 "AI: Generate Summary"

 "HTTP Request"
 "Function1"
 "Node"
 "Edit Fields"
```

Error Handling Standards

```
EVERY WORKFLOW MUST HAVE:

1. TRY/CATCH PATTERNS
   - Wrap external calls
   - Catch and handle errors
   - Don't let errors fail silently

2. FALLBACK LOGIC
   - What happens when X fails?
   - Graceful degradation

3. TIMEOUT HANDLING
   - Set reasonable timeouts
   - Handle timeout scenarios

4. ERROR NOTIFICATIONS
   - Alert on critical failures
   - Log all errors

5. INPUT VALIDATION
   - Check required fields
   - Validate data types
   - Handle unexpected input
```

Code in Code Nodes

```
// CODING STANDARDS:

// 1. ALWAYS ADD COMMENTS
// Explain what complex code does

// 2. USE DESCRIPTIVE VARIABLE NAMES
const customerEmail = items[0].json.email;  //
const x = items[0].json.email;              //

// 3. HANDLE ERRORS
try {
  // Your code
} catch (error) {
  // Handle gracefully
}

// 4. VALIDATE INPUT
if (!items[0].json.email) {
  throw new Error('Email is required');
}

// 5. KEEP IT SIMPLE
// If it's getting complex, consider breaking into multiple nodes
```

Building Workflows

Step-by-Step Process

```
PHASE 1: SETUP
─────────────
 Access n8n environment
 Verify credentials working
 Review architecture doc
 Understand requirements

PHASE 2: BUILD TRIGGER
──────────────────────
 Configure trigger node
 Test trigger works
 Validate incoming data
 Document trigger setup

PHASE 3: BUILD CORE LOGIC
─────────────────────────
 Build step by step
 Test each section
 Add error handling
 Label each node

PHASE 4: BUILD AI COMPONENTS
────────────────────────────
 Configure AI node
 Write/refine prompt
 Test outputs
 Add output parsing
 Handle failures

PHASE 5: BUILD OUTPUTS
──────────────────────
 Configure output actions
 Test end-to-end
 Verify side effects
 Clean up

PHASE 6: HARDEN
───────────────
 Add all error handling
 Add logging
 Add notifications
 Final cleanup
```

AI Development Standards

```
AI NODE REQUIREMENTS:

1. PROMPT STRUCTURE
   - System prompt first
   - Clear instructions
   - Format specifications
   - Examples if helpful

2. OUTPUT PARSING
   - Parse structured output
   - Handle unexpected formats
   - Fallback for failures

3. GUARDRAILS
   - Safety instructions
   - Topic restrictions
   - Output validation

4. TESTING
   - Test with 20+ inputs
   - Test edge cases
   - Test adversarial inputs
   - Document quality scores
```

Testing Requirements

Self-Testing Checklist

```
BEFORE SUBMITTING FOR QA:

FUNCTIONAL:
 All paths tested
 Expected outputs verified
 Side effects confirmed

EDGE CASES:
 Empty input
 Missing fields
 Very long input
 Special characters
 Unexpected data types

ERROR HANDLING:
 API failure simulated
 Timeout tested
 Invalid data tested
 Rate limit tested

AI (If Applicable):
 20+ samples tested
 Quality scores logged
 Edge cases tested
 Injection tested
```

Test Log Template

```
# Test Log - [Workflow Name]

## Test Date: [Date]
## Tester: [Name]

## Test Cases

| # | Input | Expected | Actual | Pass/Fail | Notes |
|---|-------|----------|--------|-----------|-------|
| 1 | [Input summary] | [Expected] | [Actual] | / | |
| 2 | | | | | |
| 3 | | | | | |

## Summary
- Total tests: X
- Passed: X
- Failed: X

## Issues Found
1. [Issue description + fix]
2. [Issue description + fix]

## Ready for QA:  Yes  No
```

Documentation Requirements

In-Workflow Documentation

```
STICKY NOTES TO ADD:

1. OVERVIEW (At start)
   "This workflow does X when Y triggers.
   It connects Z and outputs W."

2. SECTION HEADERS
   "==== PROCESSING ====
   Transforms incoming data for CRM"

3. COMPLEX LOGIC
   "We check X first because Y,
   then do Z if condition met"

4. IMPORTANT NOTES
   " Rate limited to 100/min
   May need adjustment for high volume"
```

Technical Documentation

```
# [Workflow Name] - Technical Documentation

## Purpose
[What this workflow does]

## Trigger
- Type: [Webhook/Schedule/etc.]
- Details: [Specifics]

## Data Flow
1. [Step 1]
2. [Step 2]
3. [Step 3]

## Integrations
| Service | Credential | Purpose |
|---------|------------|---------|
| [Service] | [Name] | [What it does] |

## Error Handling
- [How errors are handled]

## Logging
- Execution log: [Where]
- Error log: [Where]

## Known Limitations
- [Limitation 1]
- [Limitation 2]

## Maintenance Notes
- [What might need updating]
```

Communication

Daily Standup Format

```
REPORT TO TECH LEAD:

YESTERDAY:
- [What you completed]

TODAY:
- [What you're working on]

BLOCKERS:
- [Any blockers] or "None"

QUESTIONS:
- [Any decisions needed]
```

Asking for Help

```
WHEN STUCK:

1. TRY FIRST (15-30 min)
   - Check documentation
   - Search for solutions
   - Try different approaches

2. DOCUMENT THE ISSUE
   - What you're trying to do
   - What you've tried
   - What's happening

3. ASK CLEARLY
   "I'm trying to [X].
   I've tried [Y] and [Z].
   I'm getting [error/result].
   I think the issue might be [theory]."

4. SHARE CONTEXT
   - Screenshot/screen share
   - Execution ID
   - Error message
```

Reporting Issues

```
ISSUE REPORT FORMAT:

ISSUE: [Brief description]

SEVERITY: [Critical/High/Medium/Low]

DETAILS:
- Workflow: [Name]
- Node: [Where issue is]
- Expected: [What should happen]
- Actual: [What's happening]
- Error: [If any]

STEPS TO REPRODUCE:
1. [Step 1]
2. [Step 2]

ATTEMPTED SOLUTIONS:
- [What you tried]

SCREENSHOT/RECORDING: [Attach]
```

Handoff to QA

Pre-QA Checklist

```
BEFORE REQUESTING QA:

 All requirements met
 All nodes named properly
 Sticky notes added
 Error handling in place
 Logging configured
 Self-testing complete
 Test log prepared
 Documentation updated
 No test data remaining
 Production credentials ready (if applicable)

QA REQUEST:

"Ready for QA review.

Workflow: [Link/Name]
Test log: [Link]
Notes: [Anything important]"
```

Best Practices

Development Best Practices

```
DO:
 Build incrementally, test often
 Name everything clearly
 Add comments and notes
 Handle all error scenarios
 Log important events
 Keep solutions simple
 Ask questions early
 Document as you go

DON'T:
 Build everything then test
 Use default node names
 Ignore edge cases
 Hardcode values
 Skip error handling
 Over-engineer
 Struggle alone for hours
 Leave undocumented code
```

Efficiency Tips

```
WORK SMARTER:

1. USE TEMPLATES
   - Save common patterns
   - Reuse proven solutions

2. TEST INCREMENTALLY
   - Test each section
   - Don't wait until the end

3. DOCUMENT AS YOU BUILD
   - Add notes while fresh
   - Don't leave for later

4. TIMEBOX PROBLEMS
   - 30 min max stuck alone
   - Then ask for help

5. LEARN THE SHORTCUTS
   - n8n keyboard shortcuts
   - Duplicate nodes
   - Copy/paste between workflows
```

Common Patterns

Error Handling Pattern

```
[Node that might fail]
    ↓
[Error Trigger] ── [Log Error] ── [Notify Team]
    ↓
[Continue if success]
```

Logging Pattern

```
[Start of workflow]
    ↓
[Log Start] ── [Google Sheet: Start entry]
    ↓
[Main processing]
    ↓
[Log End] ── [Google Sheet: Update entry]
```

AI Processing Pattern

```
[Prepare Prompt]
    ↓
[AI Node (with retry)]
    ↓
[Parse Response]
    ↓
[Validate Output]
    ↓
[Use Output] or [Fallback]
```

Metrics

Personal Tracking

```
TRACK WEEKLY:

 Tasks completed
 Bugs introduced/fixed
 QA rejection rate
 Estimation accuracy
 Help requests given/received
```

Next: See `07-sop-client.md` for Client procedures.
