ubFlow

Agentic Workflows for Regulated Projects

Engineers decide. AI delivers.

The Pressure Is Real

Development teams face two opposing forces:

- **AI coding tools are here** — GitHub Copilot, Cursor, and others are actively used by engineers today
- **Standards don't care** — ASPICE, ISO 26262, DO-178C, IEC 62304 require the same evidence regardless of who — or what — created the artifact

> The question is not *whether* to use AI.
> The question is *how* to use it **without losing compliance**.

What Standards Actually Require

Every safety-relevant process step must be:

Obligation	ASPICE	ISO 26262	DO-178C
Bi-directional traceability	SWE.1–SWE.6	Part 8 §6	§11
Peer review with disposition	SWE.3/SWE.4	Part 6 §8	§11.9
Configuration baseline	CHG.1, SUP.10	Part 8 §7	§7
Tool qualification	SUP.8	Part 8 §11	§12
Audit trail / change records	SUP.8	Part 8 §6	§11

These are **not optional** for ASIL-B+ or DAL-C+ software.

The AI Compliance Gap

Most AI tools in use today produce **output without provenance**:

- Which instruction set was active?
- Which requirement does this code implement?
- Who reviewed it, and against which artifact version?
- Is the tool qualified for use in this project?

> Without answers to these questions, AI-generated artifacts **cannot enter a safety baseline**.

ubFlow Closes That Gap

ubFlow is a needs-driven agent design system for GitHub Copilot.

It makes AI agents natively compliant with regulated development processes — not as an afterthought, but by design:

- ✅ Every agent rule is **traceable** to a documented requirement
- ✅ Every agent action is **auditable** — full execution record per run
- ✅ Every output passes a **review gate** before entering the baseline
- ✅ Domain knowledge is **packaged and versioned** as installable agent families

How It Works: Traceable Instructions

In ubFlow, agent instructions are **not** hardcoded in config files.

They live in the project's **Sphinx-Needs documentation** — alongside requirements and specifications — and are fetched live via the **ubCode MCP server**:

```
Requirements → Specs → Flow Insts → Agent behavior
```

- ASPICE assessors can trace *why* the agent does what it does
- Changing a requirement automatically invalidates the linked instruction
- The agent's decision basis is **version-controlled and reviewable**

How It Works: Agent Families

ubFlow packages domain knowledge as **installable agent families**.

A single command installs everything needed for a domain into the project:

- Agent definitions (`.agent.md`)
- Instructions and skills (Sphinx-Needs objects)
- Process artifact templates

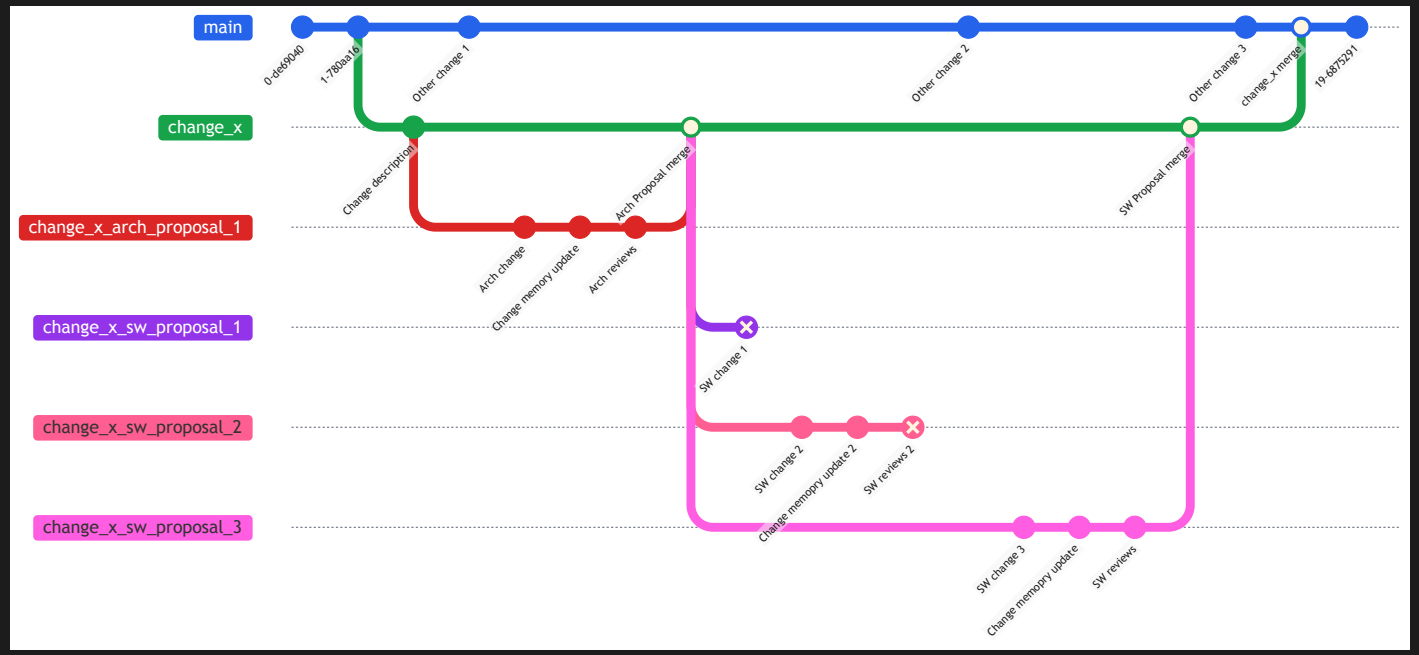
Family	Maps to
ASPICE	SYS.2, SWE.1–SWE.6, SUP.8/10 process areas
ISO 26262	Part 3 / Part 6 activities
Agile	Sprint-based iteration with traceable stories

Each family is **independently customizable** per project.

How It Works: Change Branches

AI proposals are **never merged directly** into the controlled baseline.

ubFlow uses Git branches to isolate each agentic run:



> Implements ASPICE CHG.1 and ISO 26262-8 §7 — parallel exploration without baseline pollution.

How It Works: Audit Trail

Every agentic run produces a **full execution record**:

- Prompt and active instruction set at invocation time
- All tool calls with inputs and outputs (MCP queries, file reads, API calls)
- Every artifact created or modified

This satisfies the evidence requirements for:

- **ASPICE SUP.8** — tool qualification base data
- **ISO 26262-8 §11** — software tool qualification
- **DO-178C §12** — tool qualification and usage record

Collected automatically via VS Code Copilot OpenTelemetry — **zero instrumentation effort**.

How It Works: Structured Reviews

Reviews in ubFlow are **Sphinx-Needs objects**, not free-form comments.

Each review is cryptographically linked to the exact artifact version via content hash:

```
.. review:: Architecture Review
   :by: Quality Agent
   :for: SPEC_SWE_042[hash=A3F9C1]
   :status: change_required

   Interface definition is missing error return codes.
```

- Machine-generated, human-approved, standards-compliant
- Replaces manual review documents with **structured, queryable evidence**
- Satisfies ASPICE SWE.3/SWE.4, ISO 26262-6 §8, DO-178C §11.9

How It Works: Quality Assurance

Before any artifact enters a review, the **ubFlow quality agent** evaluates it:

Category	Checks
Clarity & Wording	Ambiguous language, passive voice
Atomicity	Single-concern per artifact
Measurability	Quantified acceptance criteria
Testability	Verifiability of the stated condition
Solution Neutrality	Design leaked into requirements
Safety / Risk	Missing degradation conditions (ADAS context)

Each category yields a **score + improvement proposal**. Configurable per need type and project.

How It Works: Optimized Workflows

Every workflow step is **gate-driven**:

```
Plan → [Human approval] → Execute → [Review] → Merge
```

- The reviewed output of step N is the **controlled input** to step N+1
- Reviews are performed in **parallel** by human engineers and AI reviewer agents
- No step can proceed without explicit sign-off

> This mirrors the V-model phase gates of ASPICE and ISO 26262 — but with AI doing the heavy lifting between gates.

The ubFlow Documentation Model

All agent artifacts map directly to standard-required work products:

ubFlow Type	Standard Work Product
Flow Story (`FLST_`)	Stakeholder needs — ASPICE SYS.1 / ISO 26262-3
Flow Spec (`FLSP_`)	Software requirements — ASPICE SWE.1 / ISO 26262-6 §6
Flow Inst (`FLIN_`)	Work instructions — ASPICE SUP.8 / ISO 26262-8 §11
Flow Skill (`FLSK_`)	Process knowledge — ASPICE SUP.8
Flow Tool (`FLTL_`)	Tool qualification record — ASPICE SUP.8 BP6

Nothing outside the documentation model. **Every artifact is traceable.**

What You Get

> AI that behaves like a **qualified engineer** on your team — not a black box.

- **For managers:** Demonstrate process compliance to assessors and auditors without additional documentation effort
- **For team leads:** Domain-specific agent families that follow your process, not a generic chatbot
- **For the project:** An audit trail that satisfies ASPICE, ISO 26262, DO-178C, and IEC 62304 out of the box

Get Started

1. Install ubFlow

```
@ubFlow install the ASPICE agent family into this project
```

2. Use your agents

```
@ASPICE-Orchestrator implement change request CR-042
```

3. Review, approve, and merge

The audit trail, traceability, and review evidence are generated automatically.

📖 **Documentation:** ubflow.useblocks.com
🔧 **Source:** github.com/useblocks/ubflow