```
from typing import Callable, List, Optional, Union


from swarms.structs.agent import Agent
from swarms.utils.loguru_logger import initialize_logger


logger = initialize_logger(log_folder="swarm_reliability_checks")



def reliability_check(
    agents: List[Union[Agent, Callable]],
    max_loops: int,
    name: Optional[str] = None,
    description: Optional[str] = None,
    flow: Optional[str] = None,
) -> None:
    """
    Performs reliability checks on swarm configuration parameters.

    Args:
        agents: List of Agent objects or callables that will be executed
        max_loops: Maximum number of execution loops
        name: Name identifier for the swarm
        description: Description of the swarm's purpose

    Raises:
        ValueError: If any parameters fail validation checks
    TypeError: If parameters are of incorrect type
"""
logger.info("Initializing swarm reliability checks")

# Type checking
if not isinstance(agents, list):
    raise TypeError("agents parameter must be a list")

if not isinstance(max_loops, int):
    raise TypeError("max_loops must be an integer")

# Validate agents
if not agents:
    raise ValueError("Agents list cannot be empty")

for i, agent in enumerate(agents):
    if not isinstance(agent, (Agent, Callable)):
        raise TypeError(
            f"Agent at index {i} must be an Agent instance or Callable"
        )

# Validate max_loops
if max_loops <= 0:
    raise ValueError("max_loops must be greater than 0")

if max_loops > 1000:
    logger.warning(
        "Large max_loops value detected. This may impact performance."
    )

# Validate name
if name is None:
    raise ValueError("name parameter is required")
if not isinstance(name, str):
    raise TypeError("name must be a string")
if len(name.strip()) == 0:
    raise ValueError("name cannot be empty or just whitespace")

# Validate description
if description is None:
    raise ValueError("description parameter is required")
if not isinstance(description, str):
    raise TypeError("description must be a string")
if len(description.strip()) == 0:
    raise ValueError(
        "description cannot be empty or just whitespace"
    )

# Validate flow
if flow is None:
    raise ValueError("flow parameter is required")
if not isinstance(flow, str):
    raise TypeError("flow must be a string")


logger.info("All reliability checks passed successfully")
```