

★★★★ Day 119 ★★★★★  
of Our Daily UiPath Q&A  
Challenge!

Welcome to Day 119 of our  
exciting UiPath Q&A Challenge!  
Each day, we'll be answering one  
key question to help you master  
UiPath and revolutionize your  
automation journey. 🚀

🚫 Question for Today

What are the best practices for  
scaling REFramework?

Scaling the Robotic Enterprise Framework (REFramework) in UiPath for enterprise-grade automation involves several best practices to ensure robustness, maintainability, and performance across multiple processes and teams. Here are the key best practices:

### **1. Modular Design**

- Decouple logic: Separate business logic from framework logic using libraries and reusable components.
- Use Orchestrator assets and queues: Avoid hardcoding; use assets for configuration and queues for transaction handling.

### **2. Reusability and Libraries**

- Create shared libraries for common tasks (e.g., logging, exception handling, email notifications).
- Use invokable workflows for repetitive tasks like login, data validation, etc.

### **3. Robust Error Handling**

- Implement granular exception handling in Process.xaml and other invoked workflows.
- Use Retry mechanisms for transient errors (e.g., network issues, UI delays).
- Log exceptions with detailed context (e.g., transaction ID, activity name).

### **4. Logging and Monitoring**

- Use structured logging (e.g., JSON format) for better integration with monitoring tools.
- Leverage Orchestrator alerts and custom dashboards for real-time monitoring.

## **5. Scalability and Parallelism**

- Use multiple bots and Orchestrator queues to process transactions in parallel.
- Design workflows to be stateless and idempotent to support horizontal scaling.
- Use queue triggers to dynamically scale based on workload.

## **6. Security and Governance**

- Use Credential Assets for secure credential management.
- Implement role-based access control (RBAC) in Orchestrator.
- Audit logs and enforce code reviews and version control (e.g., Git).

## **7. Testing and CI/CD**

- Use UiPath Test Suite for unit and integration testing.
- Integrate with CI/CD pipelines (e.g., Azure DevOps, Jenkins) for automated deployment.
- Maintain versioned packages and rollback strategies.

## **8. Documentation and Standards**

- Maintain detailed documentation for each process and component.
- Follow naming conventions, code commenting, and workflow annotations.
- Use Process Design Documents (PDDs) and Solution Design Documents (SDDs).