

Building Bridges to Efficiency and Excellence.

Best Practices for Successful
UiPath Automation!

Table of Contents

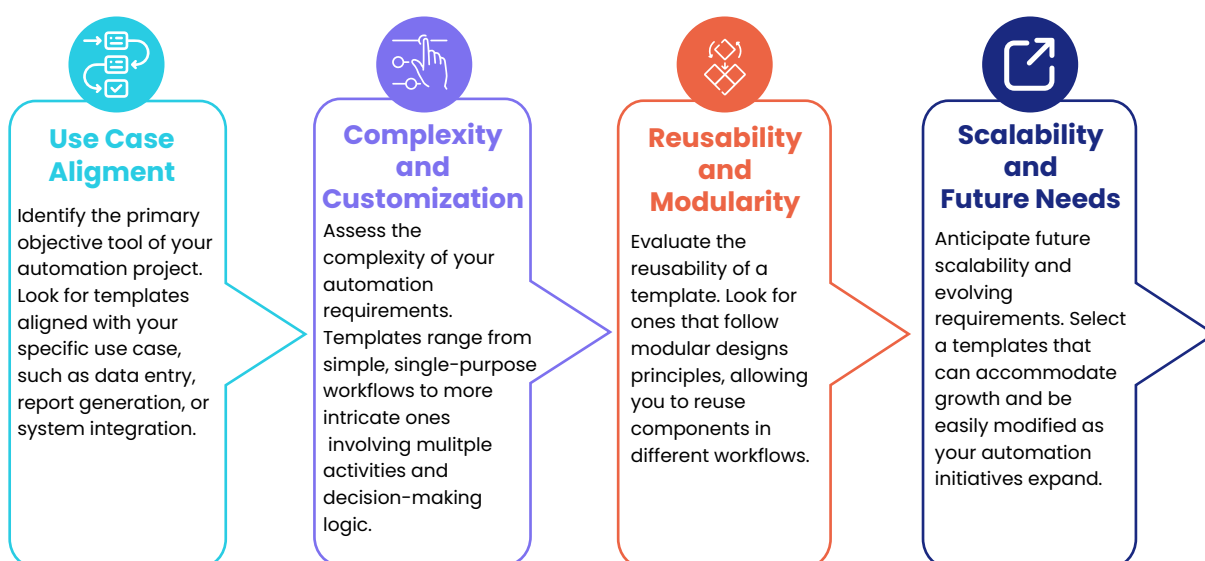
Selecting the Right Template for UiPath Bot Development	1
Project Details: Empowering Clarity and Collaboration	2
Selecting the Right Workflow Design	2-3
Creating Variables and Arguments: Improving Readability, Maintainability, and Efficiency	4-6
The Importance of Organizing Project Panel	7
The Importance of Logging in UiPath Automation	7-8
Exception Handling in UiPath Automation: How to Handle Errors and Prevent Failures	9-10
Email Notifications Improving Efficiency and Responsiveness	10-11
Power of Hybrid Automation	12
Removing Unused Elements for Efficiency and Maintainability	13
Key Benefits	13
Is there a single way to identify errors?	14-15
Effective Source Code Maintenance	16-17
Optimizing Post-Deployment Success: Documenting Best Practices for Error-Free Bot Operations	18-20
About Author	21

In today's complex business landscape, customers play a critical role in generating revenue, shaping a company's reputation, and driving innovation. Their valuable insights and long-term relationships contribute to growth and success. To prioritize customer satisfaction and meet evolving needs, businesses must deliver efficient, error-free services.

At Opteamix, we focus on delivering exceptional Robotic Process Automation (RPA) services using UiPath, an industry leader in Process Automation. By implementing best practices, we ensure our clients leverage customized RPA solutions that meet their specific requirements. This whitepaper will explore key best practices that enable us to provide reliable and efficient solutions that help organizations achieve efficiency and excellence. Our goal is to empower clients to fully leverage the potential of RPA in their business processes and achieve their business objectives.

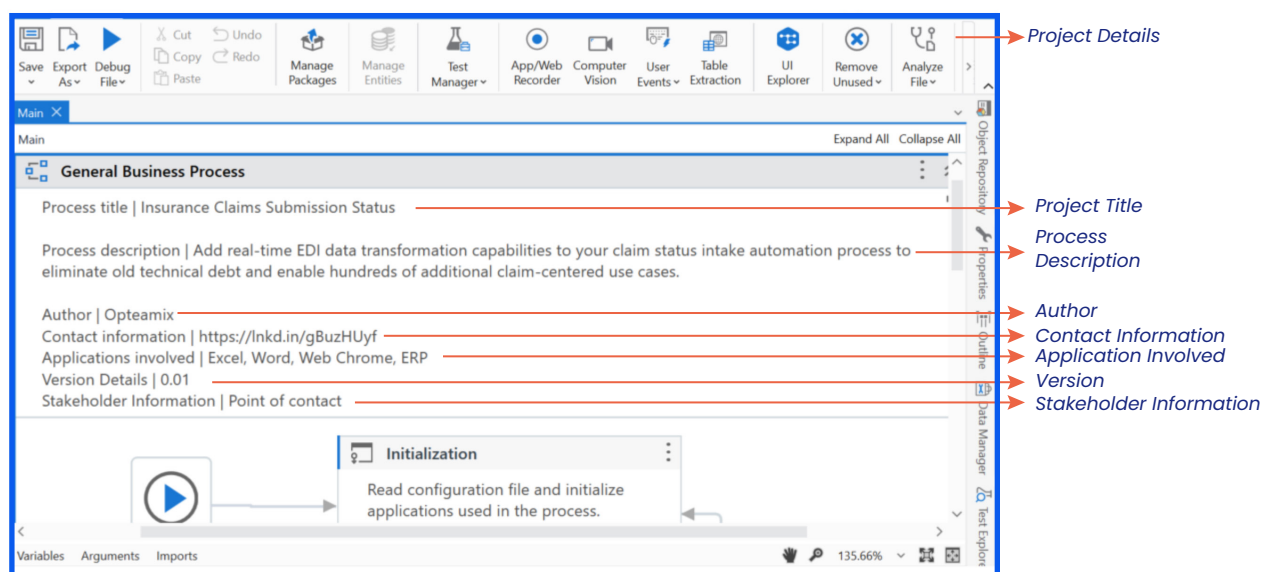
Selecting the Right Template for UiPath Bot Development

Selecting the right template is a critical first step in UiPath bot development. Templates provide a foundation and structure for building automation workflows. They offer pre-designed activities and workflows tailored to specific use cases, which can help developers accelerate development and ensure best practices.



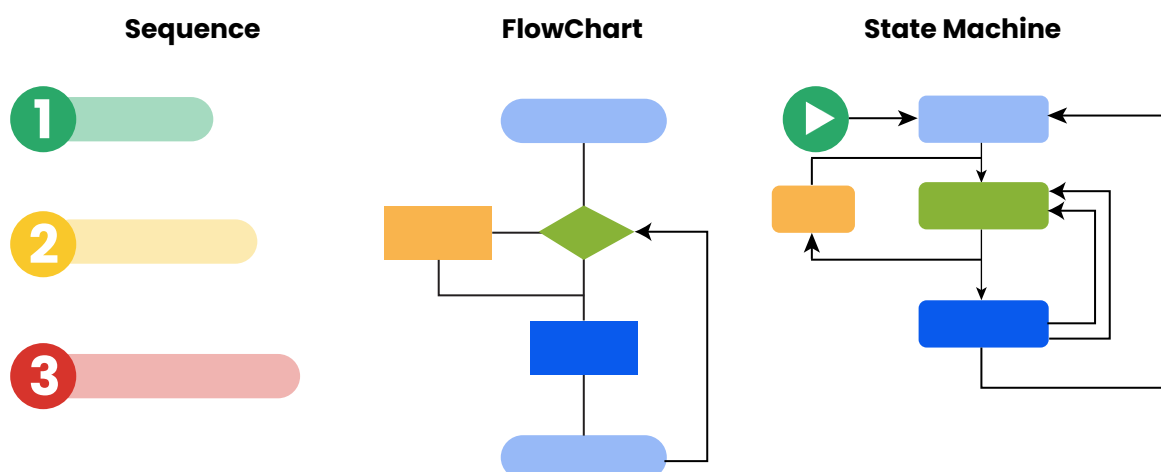
Project Details: Empowering Clarity and Collaboration

On the project main page of UiPath, it is essential to provide project details for clarity, collaboration, and effective management. These details summarize the project's purpose, ownership, and key information, facilitating better organization and communication among team members.

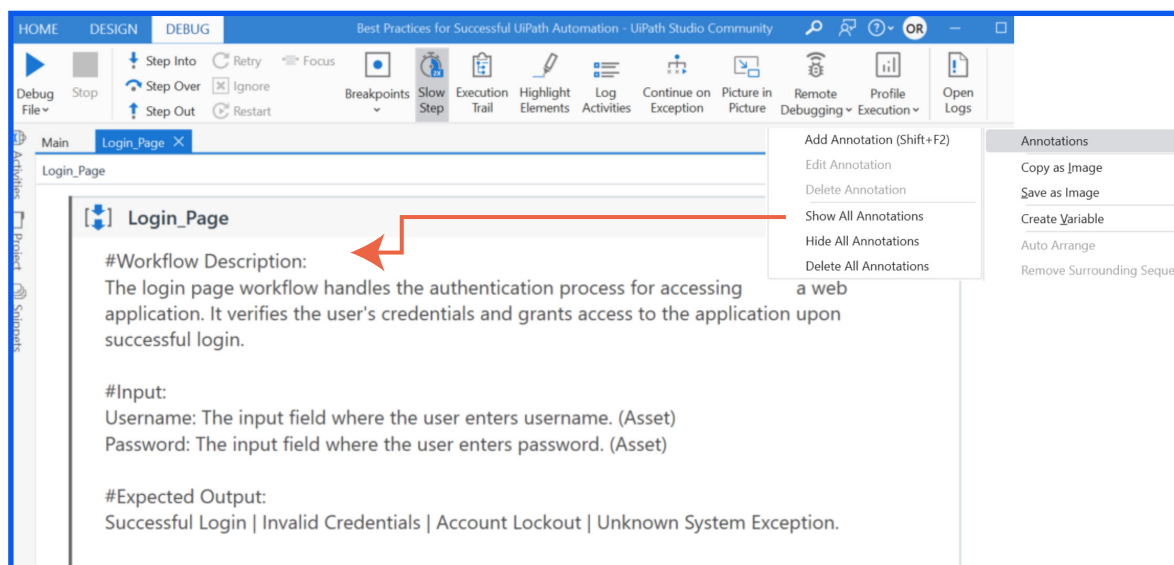


Selecting the Right Workflow Design

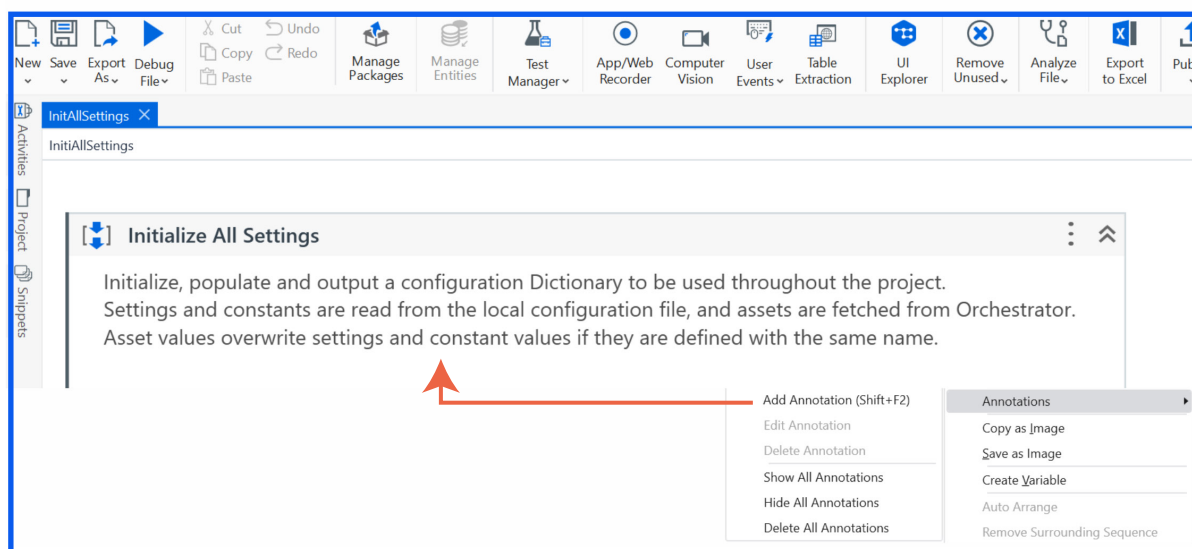
When developing automation solutions, choosing the right workflow design is essential. Three commonly used options are sequence, flowchart, and state machine.



Including clear and concise descriptions of the workflows:



By including detailed descriptions of the workflow's purpose, inputs, and expected output, developers and stakeholders gain a clear understanding of the workflow's functionality and objectives.



This clarity ensures effective collaboration, aligns expectations and facilitates the seamless development and implementation of the automation solution.

Creating Variables and Arguments: Improving Readability, Maintainability, and Efficiency

First, using descriptive variable names that indicate their purpose or content is crucial. This enhances code readability and understanding for team members.

Choosing the appropriate data type for each variable is essential to ensure data integrity and avoid unnecessary errors. Additionally, defining variables with the right scope – local or global, based on their usage helps maintain a well-organized code structure.

Initializing variables with default values is vital to establish a valid starting state and prevent unexpected behavior or null reference errors. Minimizing the scope of variables reduces memory usage and potential conflicts with similarly named variables in different parts of the code.

Instead of hard-coding numeric values, assigning them to variables with descriptive names promotes maintainability and ease of updates. Consistency in casing conventions for variable names across the project enhances code consistency and improves readability.

The diagram illustrates a table of variables with various annotations. Red arrows point from descriptive labels to specific parts of the table:

- Consistent Casing** and **Descriptive Naming** point to the 'Name' column.
- Data Type Selection** points to the 'Variable type' column.
- Default values** points to the 'Default' column.
- Scope Management** points to the 'Scope' column.
- Limit Variable Scope** points to the 'Scope' column.
- Comments** points to the 'Default' column.
- Error Handling** points to the 'Default' column.

Name	Variable type	Scope	Default
TransactionItem	QueueItem	General Business Process	Enter a VB expression
SystemException	Exception	General Business Process	Enter a VB expression
BusinessException	BusinessRuleException	General Business Process	Enter a VB expression
TransactionNumber	Int32	General Business Process	1
Config	Dictionary<String,Object>	General Business Process	Enter a VB expression
RetryNumber	Int32	General Business Process	0
TransactionField2	String	General Business Process	Enter a VB expression
TransactionField1	String	General Business Process	Enter a VB expression
TransactionID	String	General Business Process	Enter a VB expression
dt_TransactionData	DataTable	General Business Process	Enter a VB expression

Consistent Casing

Descriptive Naming

Data Type Selection

Default values

Name	Direction	Argument type	Default value
in_TransactionNumber	In	Int32	Enter a VB expression
in_Config	In	Dictionary<String,Object>	Enter a VB expression
out_TransactionItem	Out	QueueItem	Default value not supported
out_TransactionField1	Out	String	Default value not supported
out_TransactionField2	Out	String	Default value not supported
out_TransactionID	Out	String	Default value not supported
io_dt_TransactionData	In/Out	DataTable	Default value not supported

Avoiding Excessive Number of Arguments

Comments

Direction

Including comments and documentation for critical variables helps fellow developers understand their purpose and significance in the code. Utilizing variables to store error messages or status flags during exception handling enables tracking and managing issues effectively during execution. Finally, building reusable components or libraries using input and output variables to pass data between different parts of the code improves modularity and makes it easier to maintain.

It is important to define variables in a way that is generic and not specific to a particular use case. This will make the components or libraries more versatile and reusable.

a. Power of Descriptive Activity Names: Improve Readability, Maintainability, and Reusability of Your Automation Code

Renaming activities with descriptive and meaningful names in UiPath workflows has many benefits.

Desirable Activity Names

(x) Assing out_Config (initialization)

Save to: out_Config

Value to save: New Dictionary(of Stri

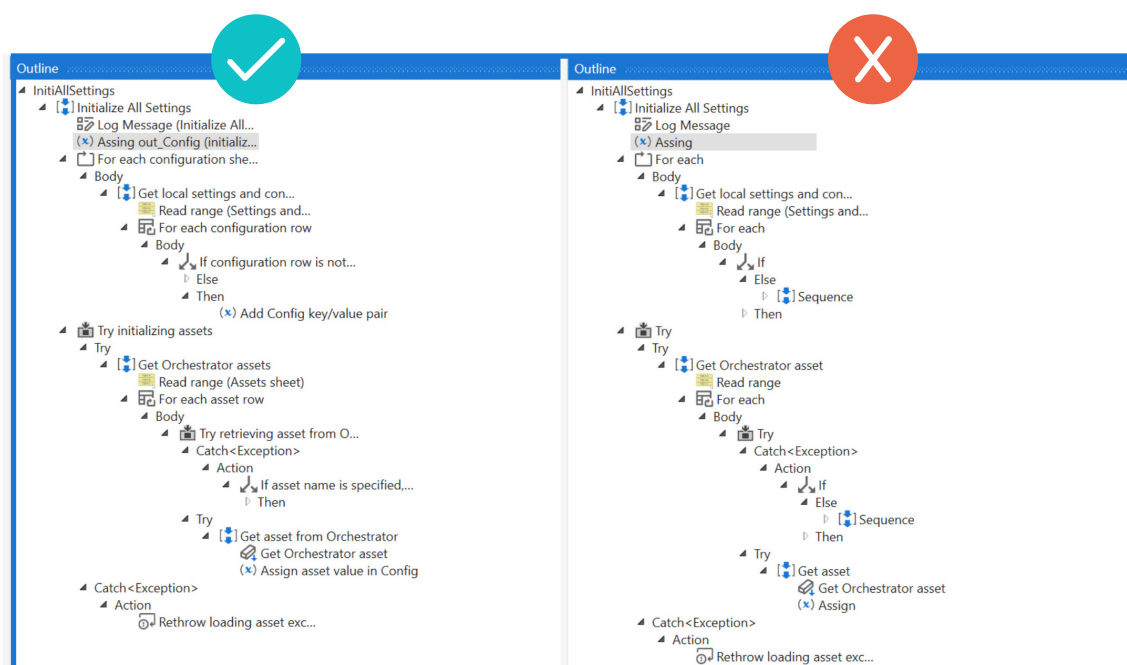
Undesirable Activity Names

(x) Assing

Save to: out_Config

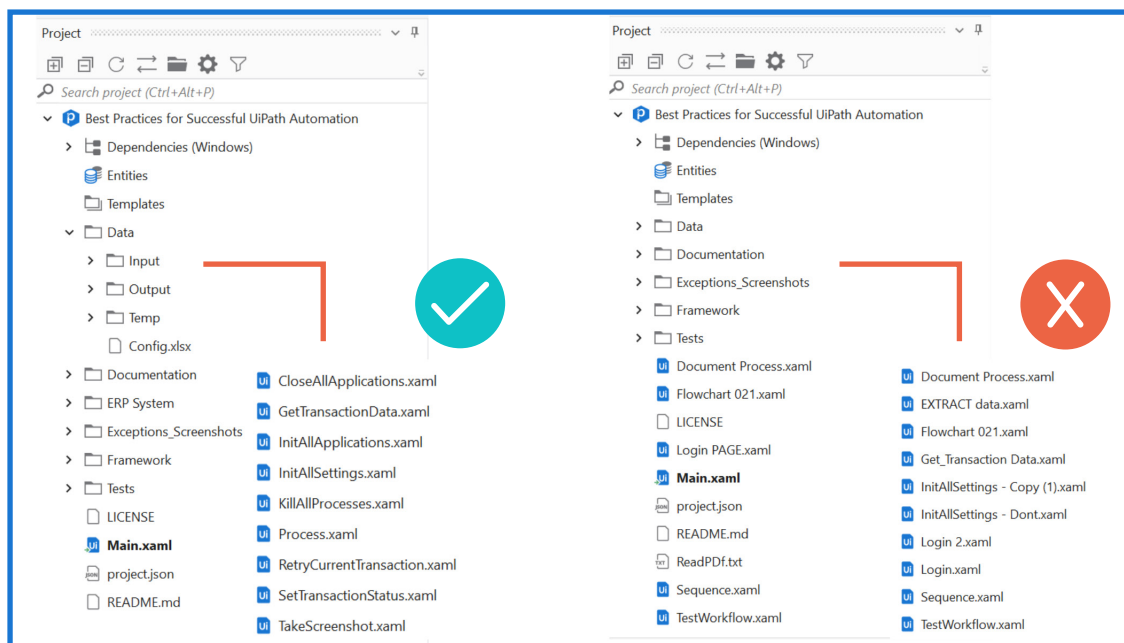
Value to save: New Dictionary(of Stri

- Renaming activities with descriptive names makes the workflow more readable, understandable, and maintainable for you and other developers.
- Descriptive activity names make it easier to maintain and update workflows over time.
- Team members can quickly understand the functionality of activities without having to spend time deciphering cryptic or unclear names.
- Descriptive activity names can help you troubleshoot issues and errors in your workflow.
- Meaningful names can provide clues about which activity might be causing the problem, leading to faster identification and resolution of issues.
- When generating workflow documentation or sharing automation processes with others, descriptive activity names can serve as a helpful reference and guide.



The Importance of Organizing Project Panel

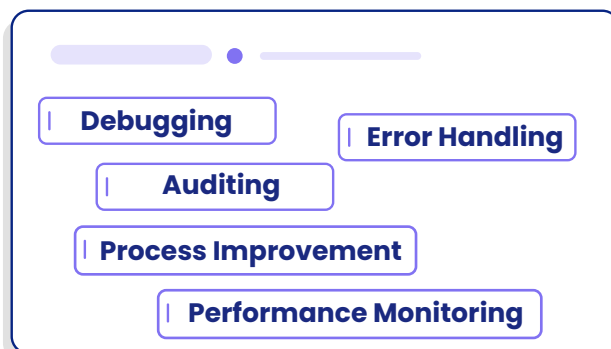
Arranging folders, files, and workflows in the project panel is essential for maintaining an organized, efficient, and reproducible automation project.



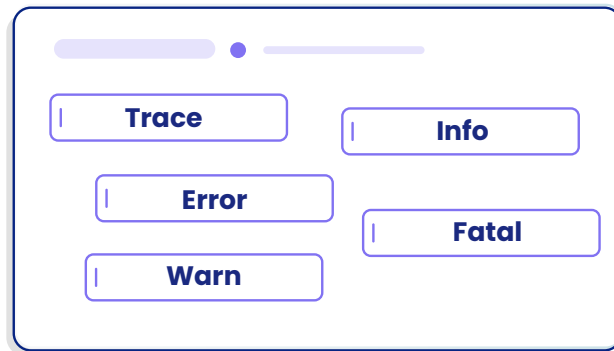
The Importance of Logging in UiPath Automation.

The log is a record of events and activities that occur during the execution of a robotic process automation (RPA) workflow. Logging is essential for monitoring and troubleshooting automation processes, as it helps to track the flow of activities, identify errors, and provide insights into the execution of the automation.

a. Importance of Log Messages



b. Types of Logs



- **Trace:** The most detailed level of logging, capturing a comprehensive record of all activities, variables, and flow control statements during the execution.
- **Warn:** Indicates potential issues or situations that may require attention but do not necessarily cause the workflow to fail.
- **Error:** Logs errors or exceptions that occur during the execution, helping to identify critical problems that must be addressed.
- **Fatal:** Represents severe errors that lead to the termination of the automation process.
- **Info:** Logs important events and general information about the workflow execution.

Exception Handling in UiPath Automation: How to Handle Errors and Prevent Failures

System Exceptions

System exceptions, or technical exceptions, are unexpected errors arising from the application or system-level failures. These errors are usually beyond the control of automation and are caused by external factors.

Business Exceptions

On the other hand, business exceptions are expected errors that occur within the application's business logic. These are situations where the application behaves as designed, but the process cannot proceed further due to specific business rules not being met.

System Exception

- **Application not responding**
- **Cannot communicate with the browser**
- **Network issues**
- **Application crashes**
- **Out of memory**
- **Element not interactable**

Business exception

- **Invalid credentials**
- **Order not eligible for discount**
- **Product out of stock**
- **Duplicate entry found**
- **Invalid format in input file**
- **Maximum transaction limit exceeded**
- **Required fields missing**
- **Customer not found**

>>

>>>

>>

>>>

c. Handling System and Business Exceptions in UiPath Automation

Exception Type	How to Handle	Purpose/Action
System Exceptions	Use the "Try Catch" activity	To gracefully handle unexpected technical errors and issues that may arise during automation. Implement error logging, retries, or alternative paths.
Business Exceptions	Use custom logic within the workflow	To address specific application-related scenarios where the process cannot proceed due to unmet business conditions. Implement decision-making, user interaction, or notification mechanisms.notification mechanisms.

Email Notifications Improving Efficiency and Responsiveness

Email notifications are a valuable feature in RPA bot development. They facilitate effective communication, issue resolution, and process monitoring. They help in maintaining a smooth automation workflow, improving collaboration between different teams, and ensuring that automation efforts align with business objectives.

An effective email about a system exception should be clear, concise, and informative. It should provide the necessary details for the recipient to understand the issue and take appropriate action.

When sending emails about system exceptions and business exceptions, you should include the following details



- **Subject**
[URGENT] System Exception Alert – [Brief Description]
- **Error Message**
The specific error message or code generated by the system.
- **Timestamp**
Date and time when the exception occurred.
- **Exception Type**
Clearly mentions types of exceptions, making it more suitable and informative for readers.
- **Process Name / Task Description**
Mention the name of the process or task being executed when the exception happened.
- **Affected System / Application**
Identify the system or application affected by the exception.
- **Exception Severity**
Indicate the severity level (low, medium, high) of the exception.
- **Environment**
Mention the environment where the exception occurred (Production, Testing, Development).
- **Impact on Process**
Briefly describe the consequences of the exception on the overall process.
- **Immediate Action Taken**
Mention any immediate actions taken to mitigate the issue.
- **Current Status**
Describe the current status of the process/task after the exception.
- **Next Steps**
Outline the proposed actions to resolve the issue and prevent recurrence.
- **Attaching Exception Screenshots**
Developers and teams quickly understand errors by showing a picture of the app when something goes wrong. This speeds up fixing issues and working together better.
- **Estimated Time for Resolution**
Provide an estimated time for issue resolution if available.
- **Contact Information**
Include contact details for relevant support teams or individuals.
- **Acknowledgment Request**
Request recipients to acknowledge the email to confirm receipt and awareness of the exception.

Power of Hybrid Automation

Build more robust, maintainable, and scalable UiPath automation solutions by leveraging the strengths of both simple logic and hybrid automation. At Opteamix, we firmly believe that embracing simplicity and clarity in design is not just beneficial for us, it would also be helpful to any potential stakeholders who may work on the automation solution in the future.

AVOIDING COMPLEX LOGICS

- *Simplicity and Readability*
- *Modularity*
- *Use Built-in Activities*
- *Limit Nesting*

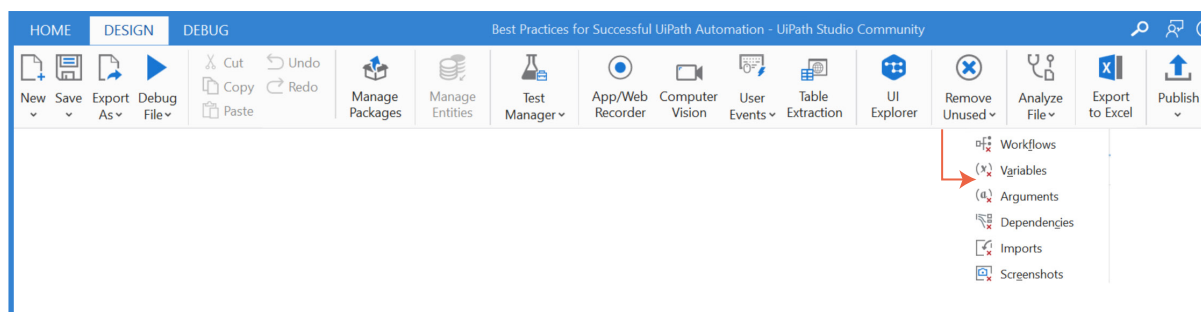
SIMPLE LOGICS TO ACHIEVE SOLUTIONS

- *Leverage UiPath Activities*
- *Avoid Over-Engineering*
- *Use Decision Tables*

HYBRID AUTOMATION

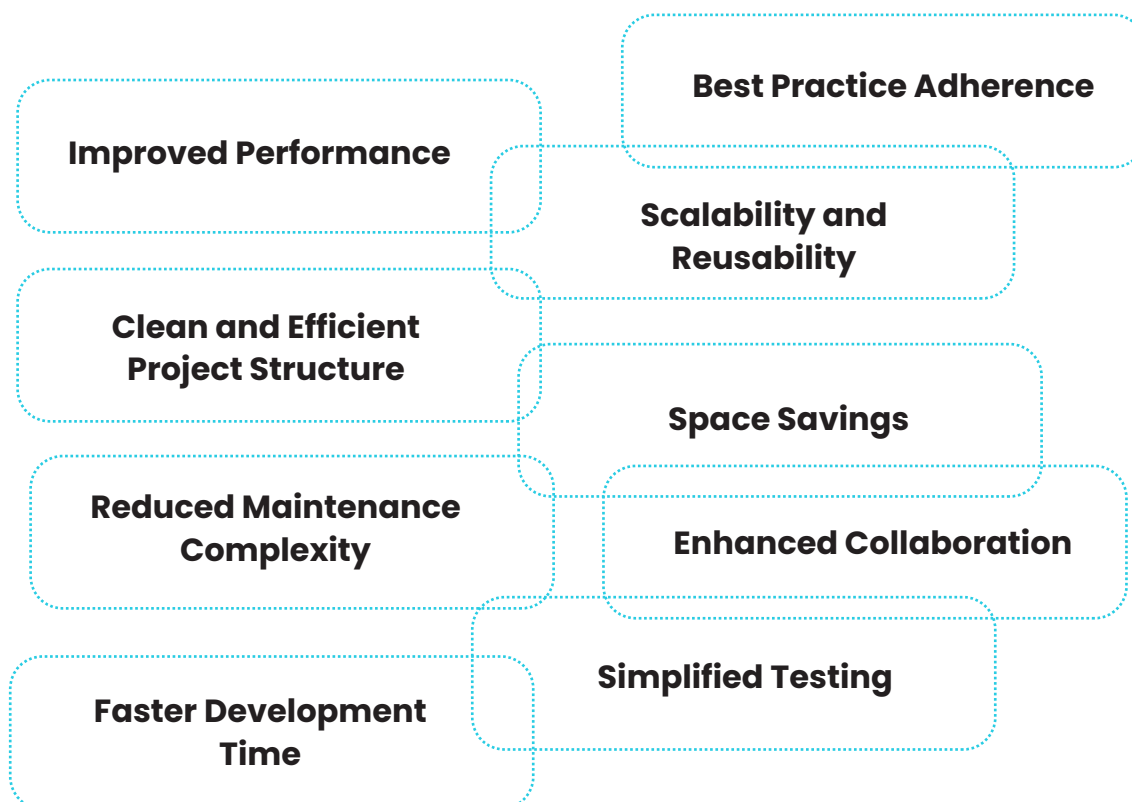
- *Combine Attended and Unattended Automation*
- *Error Handling and Human Intervention*
- *Orchestration*

Removing Unused Elements for Efficiency and Maintainability



Removing unused workflows, variables, arguments, dependencies, imports, and screenshots in UiPath development is crucial for maintaining a clean, efficient, and well-organized automation project. This practice yields benefits such as improved performance, easier maintenance, enhanced collaboration, reduced risk of errors, and a more streamlined development process.

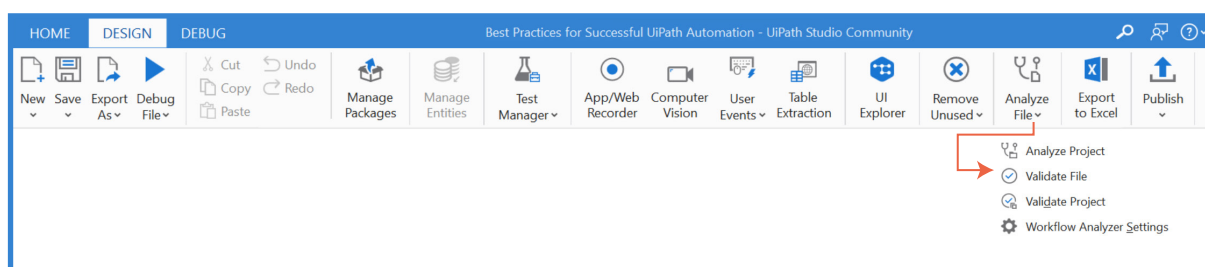
Key Benefits



Is there a single way to identify errors?

a. Workflow Analyzer

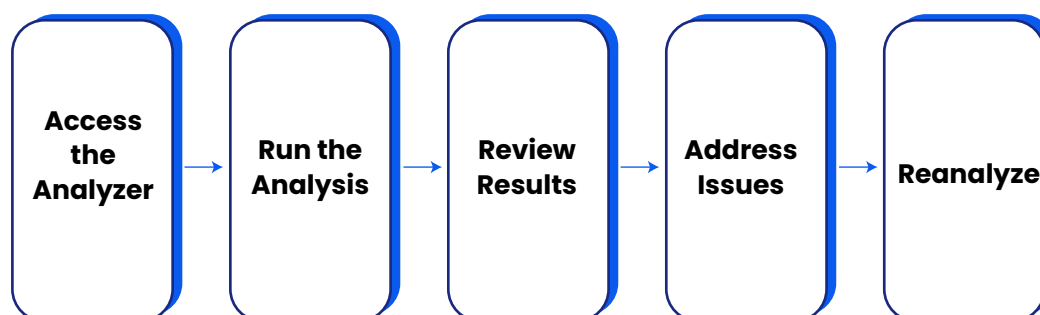
UiPath's Workflow Analyzer is a crucial tool for ensuring the quality, best practices adherence, and efficiency of your RPA workflows. Use it during development, code review, and maintenance stages to reap its full benefits.



b. When to Use Workflow Analyzer

- **Development:** You can use the Workflow Analyzer to identify and fix issues in your workflows early in the development process.
- **Code Review:** Before sharing or deploying your workflows, you can run the analyzer to ensure they meet quality and best practices standards.
- **Maintenance:** When updating and modifying existing workflows, the analyzer maintains the same level of quality and adherence to best practices.

c. How to Use Workflow Analyzer



- **Access the Analyzer:** In UiPath Studio, open the workflow you want to analyze. Go to the “Design” tab and click on “Workflow Analyzer” in the “Review” section.
- **Run the Analysis:** Click “Run Analysis” to start the analysis process. The Workflow Analyzer will inspect your workflow for issues and best practices violations.
- **Review Results:** After the analysis is complete, you will see a report detailing issues, warnings, and information about the workflow. The issues are categorized with severity levels.
- **Address Issues:** Click on each issue to see a description and recommended solution. You can make the necessary changes directly in UiPath Studio.
- **Reanalyze:** Once you have made changes, you can run the analysis again to ensure all identified issues have been resolved.

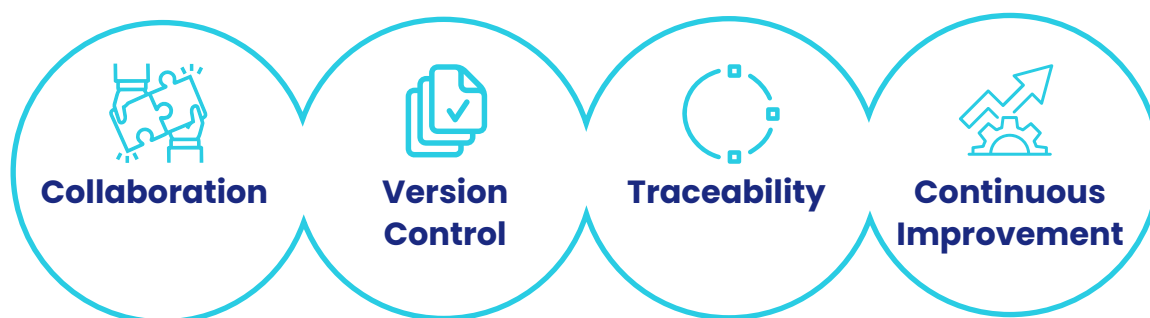
d. Benefits of Workflow Analyzer

- **Code Quality Improvement:** Workflow Analyzer helps you identify and rectify coding issues and anomalies, leading to better quality workflows.
- **Best Practices Adherence:** It ensures that you follow UiPath’s recommended best practices, enhancing the long-term maintainability of your automation solutions.
- **Performance Optimization:** The tool suggests optimizations that can improve the execution speed and efficiency of your workflows.
- **Consistency:** Workflow Analyzer ensures a consistent coding style across your team’s projects, promoting collaboration and understanding.
- **Reduced Errors:** By detecting potential issues early, Workflow Analyzer helps prevent errors and bugs in your automation processes.

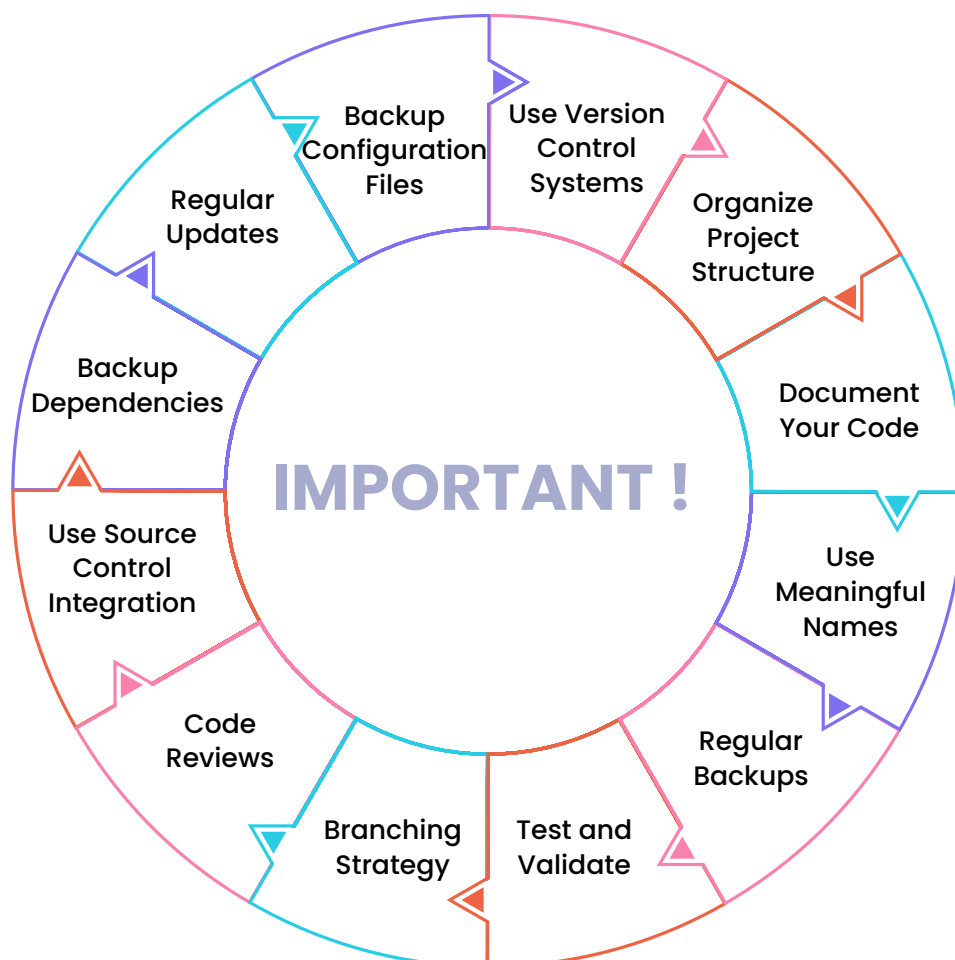
Effective Source Code Maintenance.

Remember that maintaining source code is not just about the code itself, it encompasses the entire development process and the collaboration among team members. A well-maintained project will save time, reduce errors, and lead to a more efficient development lifecycle.

a. Importance of Maintaining Source Code



b. Maintaining Source Code



- **Use Version Control Systems:** Utilize tools like Git, which provide version control features. Create a repository for your project and commit changes regularly with clear and descriptive commit messages.
- **Organize Project Structure:** Keep your project organized with a clear directory structure. Divide workflows, assets, and configurations into logical folders for better readability.
- **Document Your Code:** Add comments and documentation to explain complex parts of your code. This helps other developers understand your intent and reduces confusion.
- **Use Meaningful Names:** Give variables, activities, and files meaningful names that reflect their purpose. This improves readability and maintainability.
- **Regular Backups:** Back up your project files regularly to avoid data loss due to unexpected events.
- **Test and Validate:** Before committing changes, thoroughly test and validate your code to ensure it functions as expected.
- **Branching Strategy:** Utilize branching strategies in Git to work on new features or fixes without disrupting the main codebase. Merge the changes only after thorough testing.
- **Code Reviews:** Collaborate with team members and conduct code reviews to catch issues early and ensure coding standards are followed.
- **Use Source Control Integration:** UiPath Studio integrates with source control systems like Git, making it easier to manage your project directly from the Studio interface.
- **Backup Dependencies:** Store dependencies, libraries, and custom activities in a version-controlled repository to ensure consistency across different development environments.
- **Regular Updates:** Keep your automation environment and tools up to date to benefit from the latest features and security patches.
- **Backup Configuration Files:** Keep track of configuration files and settings separately from the source code to ensure easy migration between environments.

Optimizing Post-Deployment Success: Documenting Best Practices for Error-Free Bot Operations

After developing and deploying an UiPath bot, it is crucial to ensure client satisfaction and error-free operation. To achieve this, you should consider submitting the following documents as best practices:

User Manual

**Process
Documentation**

**Deployment
Documentation**

**Configuration
Guide**

**Error Handling
and
Troubleshooting
Guide**

**Logging
and
Monitoring**

**Data Security
and Privacy
Documentation**

**Change
Management
Plan**

**User Acceptance
Testing
(UAT) Plan**

**Backup and
Recovery
Strategy**

**Service Level
Agreement
(SLA)**

**License and
Compliance
Information**

**Contact
Information**

- **User Manual:** Prepare a detailed user manual that provides step-by-step instructions on how to use the bot. Include information about input requirements, expected outputs, and troubleshooting steps.
- **Process Documentation:** Document the entire automation process, outlining each step, decision point, and interaction the bot performs. This helps understand the process flow and identify potential areas for improvement or error handling.
- **Deployment Documentation:** Describe the deployment process, including instructions on how to install, configure, and run the bot in different environments. This documentation ensures consistent deployment across different setups.
- **Configuration Guide:** If the bot requires configuration settings or parameters, create a configuration guide explaining how to set these options for optimal performance properly.
- **Error Handling and Troubleshooting Guide:** Develop a guide that covers common errors, exceptions, and possible issues that users might encounter. Provide detailed troubleshooting steps and solutions to resolve these problems.
- **Data Security and Privacy Documentation:** If the bot interacts with sensitive data, detail how the data is handled, stored, and secured to comply with data protection regulations and client privacy requirements.
- **Change Management Plan:** Outline a plan for handling future updates, enhancements, and changes to the bot. This plan should cover version control, testing procedures, and communication strategies.
- **User Acceptance Testing (UAT) Plan:** Specify the UAT procedures to be followed by the client before the bot is considered fully accepted and ready for production use. This ensures that the bot meets the client's expectations.
- **Backup Configuration Files:** Keep track of configuration files and settings separately from the source code to ensure easy migration between environments.

- **Service Level Agreement (SLA):** Define the agreed-upon service level agreement, including response times for support requests, maintenance schedules, and performance expectations.
- **License and Compliance Information:** Provide details about the software licenses, dependencies, and any compliance requirements the bot adheres to.
- **Contact Information:** Include a list of contacts for technical support, troubleshooting, and any questions related to the bot.

In conclusion, adhering to these UiPath best practices is essential for ensuring the successful development and deployment of high-quality automation solutions. By following these guidelines, developers can pave the way for a smoother development lifecycle, enhanced reliability, and improved maintainability of their projects. Adopting these best practices as the technological landscape evolves reflects a commitment to delivering top-notch products and sets the stage for continuous innovation and growth. Let's work together to raise the bar for excellence in UiPath development and contribute to a future of efficient and sustainable automation. Your dedication to these best practices is a cornerstone of our shared success. Happy Automation! May your solutions be innovative, your workflows be smooth, and your customers be satisfied.

Author



Srenivasan Kannan
Technical Architect

Srenivasan Kannan is a Technical Architect at Opteamix and is part of the Robotic Process Automation Practice. With over a decade of formidable tenure, Srenivasan has cultivated exceptional proficiency in RPA, Smart Script Automation, Artificial Intelligence (AI), and Machine Learning (ML) solutions. Srenivasan's richly diverse background spans key industry domains, including Finance, Logistics, and IT, enabling him to seamlessly integrate his technical acumen into each sector and deliver tailored, high-impact solutions.

Srenivasan occupies a pivotal role as Opteamix's RPA Practice Lead, assuming oversight of a team of technology professionals entrusted with delivering cutting-edge automation solutions. Leveraging his extensive expertise, Srenivasan has consistently pioneered the development of intelligent RPA solutions. His nuanced approach to designing performance-enhancing methodologies has consistently yielded remarkable outcomes, underscoring his unwavering commitment to excellence. His exceptional aptitude for conceiving and executing reusable automation frameworks has engendered a paradigm shift in optimizing operational processes across various practice units.

His RPA expertise, unparalleled contributions, and profound insights have merited him the esteemed title of Most Valuable Professional (MVP) for the year 2023. This accolade eloquently underscores his profound impact and exceptional eminence in the RPA community.



Opteamix is a digital automation technology consulting firm with deep expertise in Application Development, Robotic Process Automation, AI, DevOps, Enterprise Mobility, and Test Automation Services. As a next-generation technology firm, we work on innovative technologies to automate and digitally transform our client's businesses.

From building innovative financial service platforms and managing applications that aid in administering health benefits to automating business processes, we help develop and maintain mission-critical applications vital to our client's business. Opteamix leverages Agile and SAFe methodologies to deliver projects using global teams that help us deliver positive outcomes for our clients at an optimized cost and faster time to market. We are headquartered in Denver, Colorado, with a wholly-owned delivery center in Bengaluru, India.

Get in touch

USA

9609 S. University Blvd.
#631994
Littleton
CO 80163

+1 (303) 683-6454

INDIA

Brigade Software Park
Opteamix, A Block, 2nd Floor
27th Cross Road
Banashankari Stage II
Bengaluru 560070

+91 80 4667 1666

Visit us at

www.opteamix.com

