

Technical Testing Plan Best Practices

Technical testing plan best practices

- Define the scope of testing, including in-scope and out-of-scope elements.
- Choose appropriate testing types based on the project needs (e.g., unit testing, integration testing, system testing, performance testing, compatibility testing).
- Set up a dedicated and controlled test environment that mirrors the production environment as closely as possible.
- Set up a dedicated and controlled test environment that mirrors the production environment as closely as possible.
- Specify the versions of UiPath Studio and Robot to be used for testing.
- List any additional software dependencies required for testing.
- Ensure all necessary licenses are available.
- Ensure that all necessary hardware, software, and network configurations are in place.
- Develop detailed test cases that cover functional and non-functional requirements.
- Clearly document test inputs, expected results, and execution steps.
- Include positive and negative scenarios to cover a wide range of conditions.
- Outline the sequence in which the test cases will be executed.
- Define the criteria for passing or failing a test.
- Prioritize test cases based on criticality and impact.
- Define and manage test data to cover various scenarios, including boundary conditions and edge cases.
- Ensure data privacy and security compliance, especially when dealing with sensitive information.
- Anonymize or pseudonymize sensitive data used in testing to comply with data protection regulations.
- Execute test cases systematically and monitor test progress.
- Log and report defects promptly, including detailed steps to reproduce and associated artifacts.
- Simulate realistic user scenarios and gradually increase the load to identify performance bottlenecks.
- Monitor system resources during performance testing.
- Conduct a retrospective analysis after each testing cycle to identify improvements.
- Regularly review and update test cases to reflect changes in requirements.
- Foster strong communication and collaboration between testing and development teams.
- Establish traceability between test cases, requirements, and design specifications.
- Ensure that each requirement is covered by one or more test cases.
- Conduct reviews of test cases to ensure clarity, completeness, and accuracy.
- Involve multiple team members in the review process to gain diverse perspectives.
- Include positive and negative scenarios to cover a wide range of conditions. **Avoid Unnecessary**