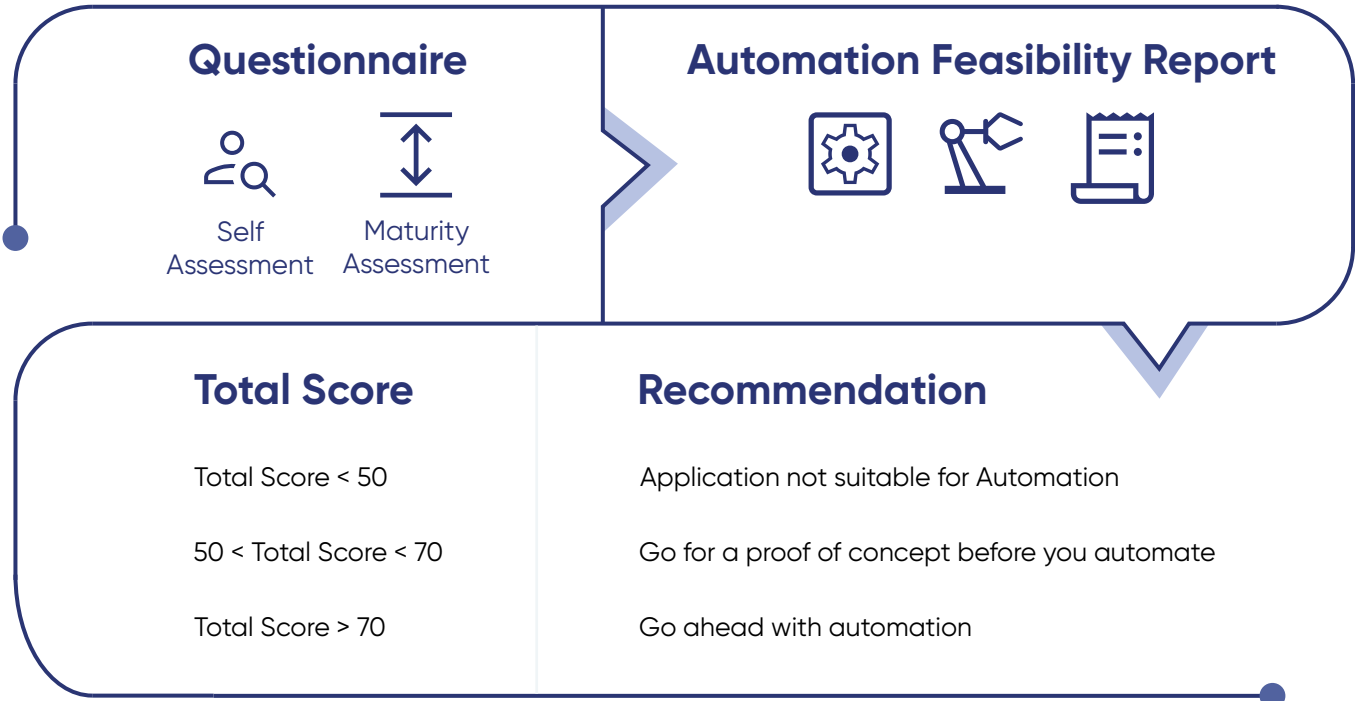


# Evaluate Your Test Strategy!

A robust testing strategy is paramount to business success. Evaluating your testing strategy at regular intervals is critical to ensure its effectiveness and alignment with your project goals. This evaluation involves assessing key areas such as team composition, technology, processes, and reporting.



## Self-Assessment

Developed to evaluate your organization's performance against industry standards by conducting a thorough analysis of different components.



### Team

- Team composition
- Expertise gaps
- Collaboration between QA and Dev teams



### Technology

- Deployment pipeline makeup
- Automation integration
- Trigger points



### Process

- Overall test process
- Speed of sprint cycles
- Feedback methods



### Reporting

- Test coverage
- Common devices used
- Automation ROI?



### Pain

- Existing pain points
- Overlooked errors

# Maturity Assessment

Effective strategies require continuous implementation, maturity evaluation, journey mapping, and balanced testing approaches.

	Chaos	Continuous Integration	Continuous Flow	Continuous Feedback	Continuous Improvement
People	<ul style="list-style-type: none"> <li>• Silo team organization</li> <li>• Little knowledge of test automation</li> <li>• Blame, finger-pointing</li> </ul>	<ul style="list-style-type: none"> <li>• Limited knowledge of testing, Ad-hoc training</li> <li>• Some Dev/QA co-ordination</li> </ul>	<ul style="list-style-type: none"> <li>• Test automation skills and training program</li> <li>• Risk management</li> <li>• Dev/QA joint plan</li> </ul>	<ul style="list-style-type: none"> <li>• Collaboration using shared test metrics</li> <li>• Goals SLI/O/As</li> <li>• Mentors and Guilds</li> </ul>	<ul style="list-style-type: none"> <li>• Experimentation</li> <li>• Integrated Dev/QA</li> <li>• E2E user experience focus</li> </ul>
Process	<ul style="list-style-type: none"> <li>• Testing not part of planning</li> <li>• No test standards</li> <li>• Few automated tests</li> </ul>	<ul style="list-style-type: none"> <li>• Most tests other than build tests are manual</li> <li>• Minimal test version management</li> </ul>	<ul style="list-style-type: none"> <li>• E2E CI/CD pipeline, test visible</li> <li>• Test/release standards</li> <li>• Test management</li> </ul>	<ul style="list-style-type: none"> <li>• E2E performance trends drive test design</li> <li>• Focus on removing test bottlenecks</li> </ul>	<ul style="list-style-type: none"> <li>• Risk based test design</li> <li>• Automated test creation and test results analysis</li> </ul>
Technology	<ul style="list-style-type: none"> <li>• Missing tools to test performance of applications, pipelines and infrastructure</li> </ul>	<ul style="list-style-type: none"> <li>• Version management</li> <li>• Automated build tests</li> <li>• Painful but repeatable releases</li> </ul>	<ul style="list-style-type: none"> <li>• Most tests automated for app, infra, pipeline</li> <li>• Release metrics use test results</li> </ul>	<ul style="list-style-type: none"> <li>• Test environment orchestration</li> <li>• Predictive test analytics</li> </ul>	<ul style="list-style-type: none"> <li>• E2E value stream test analysis, orchestration and execution</li> <li>• Intelligent test creation</li> </ul>



## Improved Quality and Performance

Identify and fix issues before release, making the product more maintainable and less error prone.



## Streamlined Product Development

Creating more streamlined products to improve efficiency and release to users faster.



## Increased Test Efficiency and Reusability

Enhance test efficiency and speed, increase test coverage, and enrich test reusability.