TARPON HEALTH

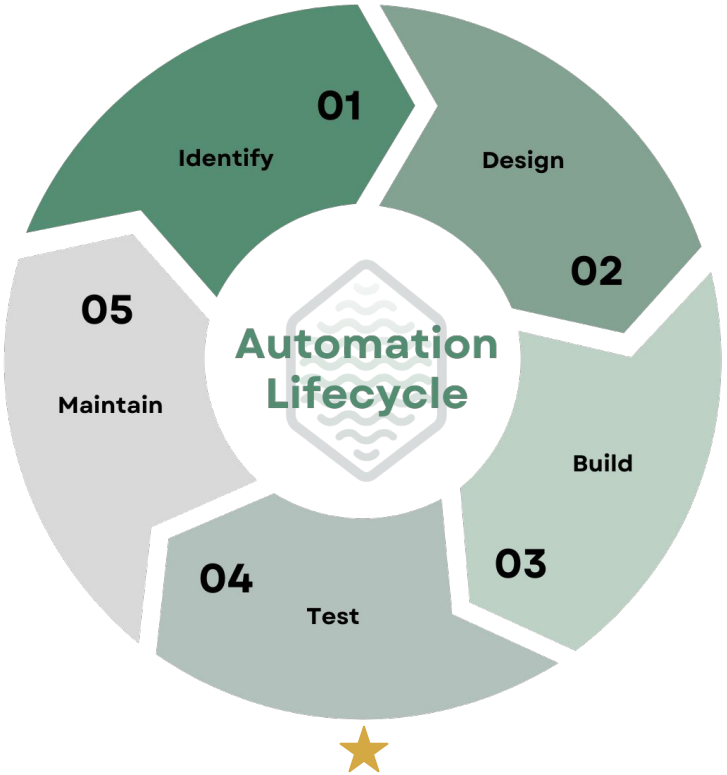
Five phases of testing

Agenda

What we will cover today

1. Unit testing
2. Integration testing
3. User acceptance testing
4. End-to-end testing
5. Production testing

The Automation Lifecycle



Types of testing

The following phases provide a framework and definitions around how to test an automation solution.

Unit testing	Integration testing	User acceptance testing	End-to-end testing	Production testing
Unit Testing is the process of testing individual units of code. This is the most basic level of testing, and its purpose is to ensure that each unit of code works correctly.	Integration Testing is the process of testing how different units of code and applications interact with each other. Testing the connections ensures that the automation solution works as a whole.	User Acceptance Testing is the process of testing the accuracy of the automation solution with users. This ensures the automation meets the needs of the business.	End-to-end Testing is the process of testing the automation solution in its entirety. This should be done in a testing environment.	Operational Testing is the act of monitoring the automation in the production environment and making adjustments until the automation runs error free.

One big caveat

While the phases are linear, **we recognize that this is not necessarily a linear process**

Unit testing

Testing starts the day you begin building.

How to do it

Unit Testing is ideally completed throughout the build phase and at the end.

Testing is typically done through peer-to-peer exercises, where one developer reviews another's work.

A more senior developer or manager should check a more junior developer's work if the resources are available.

Emphasis should be put on whether the automation is build to the proper standards.

Discussion topics

- Can someone describe their current process?
- What do you do if you only have one developer?
- How do developers check each other’s work at your organization?
- How much manager involvement is needed (should they be)?
- What is a good way to ‘trip-up’ an automation

Integration testing

Make it a point to focus on the ‘joints’ (i.e., where they connect) of your automation.

How to do it

Identify the modules, processes, and systems that must work together.

Each connection must be tested to verify that the different components interact correctly.

Focus on accuracy of the information and timing.

The system environment for development must match the production environment (e.g., access to files, resolution, etc.); any known deviations must have a plan.

Discussion topics

- What are some examples of integration failures?
- What are common system environmental issues that you encounter?
- Who is typically involved from IT in integration testing?

User acceptance testing

If the automation isn't meeting the needs of the business, then why build it at all?

How to do it

Identify the business process owners impacted by the automation.

Build a plan with a wide range of account types and testing scenarios.

Have business process owners grade the accuracy of the automation outputs.

Make code adjustments based on exceptions and failures.

Discussion topics

- What is the best way to create all the scenarios?
- What is the threshold for starting User Acceptance Testing? How fluid is this process? Does anyone leverage Test Driven Development?
- What amount of volume do you typically give to a business process owner for accuracy validation?

End-to-end testing

Let the automation run in its entirety and evaluate the outcomes.

How to do it

Run the automation from start to finish (in a testing environment where possible).

Use the designated schedule with as much volume as possible in a testing.

Watch the automation and confirm that the automation is running free of manual intervention, capturing the right volume, and producing the anticipated results.

Establish baseline metrics.

Discussion topics

- How many days or runs are enough?
- What categories of failures are you documenting?
- Do you re-check accuracy or do you assume that the accuracy is good from User acceptance testing?
- Who approves this phase of testing is complete?
- What level of success is deemed as “production ready”?

Production testing

Watch all your work come to life.

How to do it

Review security protocols and system environment differences.

Inform business process owners before making final adjustments and moving the automation into production.

Push to production.

Watch the automation as it runs in production and make any changes in concert with management.

Discussion topics

- What are common production go-live issues?
- Do you have any rules about production pushes?
- How many days do you watch the automation before you feel comfortable letting it run unmonitored?
- Who approves this phase of testing is complete?

Closing thoughts

- One of the biggest challenges that organizations will face is creating testing scenarios for user-acceptance testing.
    - Organizations should identify edge cases that have been addressed during the design and build phases and make sure they are being tested.
- Often, you will find accounts that do not fit the original workflow. Do not allow outliers or new types of accounts to lead to scope creep. Log them as enhancements for future development initiatives.
- There will always be slight differences between testing and production environments. Therefore, it is essential to perform due diligence ahead of the production deployment and carve out time to make adjustments during the first week.

Use our testing toolkit to organize Integration, User Acceptance, and End-to-end testing

What other tools and information would be helpful in this space?