POST CAPTAIN
CONSULTING

BEST PRACTICES

SLATE TESTING GUIDE

LEARN MORE postcaptain.com

INSIDE THIS GUIDE

You just built something new in Slate. Maybe it's a portal. A custom application page. A complex workflow. It looks good on your screen. You hit "save," cross your fingers, and hope for the best.

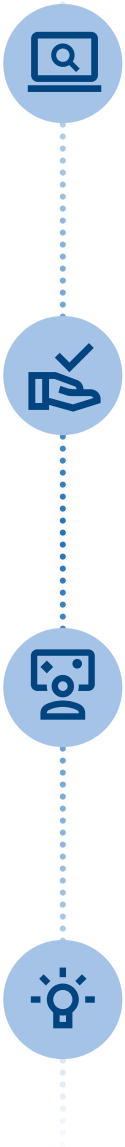
But hope is not a strategy.

That's why we created this guide: to give Slate Captains the tools and confidence to test their work properly before it goes live.

POST CAPTAIN
CONSULTING

THE *WHY* BEHIND TESTING

There's no one-size-fits-all template, and that's okay. This guide gives you the thinking tools and practical examples to develop your own repeatable testing process.



EFFICIENCY

Proactively testing your builds not only protects project quality, but also saves staff time, preserves team bandwidth, and reduces the likelihood of costly surprises after go-live. In short: a little testing now saves a lot of fixing later.

CERTAINTY

Thorough testing helps eliminate guesswork. You can move forward with certainty knowing that what you've built works as intended, handles the most likely edge cases, and won't break under normal use.

USABILITY

Testing allows you to experience your builds from the end user's point of view and ensure the experience is intuitive, accessible, and accurate. These seemingly minor details often make the biggest difference in how your community experiences your system.

OPTIMIZATION

When done well, testing generates real-time feedback and new insights. It gives you the perspective needed to refine your work before it goes live, and lays the groundwork for iterative improvements that can make your system more resilient over time.

DIFFERENT TYPES OF TESTING

Testing in Slate ensures every form, portal, automation, and integration works as intended, and that users can navigate the system with ease.

HERE'S HOW DIFFERENT TYPES OF TESTING *BUILD CONFIDENCE* AT EVERY STAGE:

FUNCTIONAL TESTING

Unit testing small pieces of logic (like form conditions, rule actions, or liquid tokens) as you build helps catch issues early and minimizes rework down the line.

Regression testing comes into play as your instance evolves, making sure that new changes don't unintentionally break existing functionality.

End-to-end testing helps you experience your entire process as a user would to confirm that data flows, automations trigger, and portals respond correctly.

USER TESTING

Acceptance testing confirms that your build meets the original requirements, often best done with the person or team who requested the feature.

Usability testing evaluates whether the process is intuitive and accessible to those using it. Are instructions clear? Do buttons and fields behave as expected?

MAKING A TEST PLAN

Before you start clicking through your build, take a step back and create a simple test plan. With a solid test plan in place, you're set up for more reliable, confident launches (and fewer surprises after go-live).

WHY YOU NEED A TEST PLAN

- **Consistency**: Everyone follows the same steps, reducing guesswork.
- **Repeatability**: You can re-test easily after making changes.
- **Documentation**: Record a clear description of what was tested and what passed.

WHAT SHOULD BE IN YOUR TEST PLAN?

Your plan doesn't need to be complicated. It can range from a simple checklist to a detailed spreadsheet, whatever fits your team and the complexity of the build. At a minimum, try to define the following for each test case:

- **What are you testing?** Be specific. (e.g., "Inquiry form auto-response email triggers with correct personalization.")
- **What's the expected result?** Clearly state what should happen when the test is performed.
- **Did it pass?** Record the outcome, and note any issues or unexpected behavior.
- **Does it need re-testing?** If you do encounter an unexpected behavior, confirm it with a follow-up test.

HOW TO THINK LIKE A TESTER

- **Include meaningful test cases:** Make sure you're covering not just functionality, but logic, data flow, and user experience.
- **Don't just test anticipated actions:** Try to break things. Enter data incorrectly. Skip steps. See what happens.
- **Balance is key:** It's smart to cover edge cases, but know when to stop. Some rare scenarios aren't worth the effort. Focus on what's most likely to happen in real use.

TEST PLAN EXAMPLES

AUTOMATED APPLICATION FEE & PAYMENT ACTIVITY TRACKING

Step	Expected Result	Notes	Pass?	Testing Notes
Create a test application	App Status = Awaiting Submission Bin = Awaiting Submission Application Created email sent	In order for the email to trigger, I think the application needs to be created as a student, not administratively. Direct round URL: https:[redacted]apply/?sr=cd6ffc22-2525-4b3b-abb9-3124274fc880	Y	
Submit an application (make sure two recommenders are added and awaiting)	App Status = Awaiting Materials Bin = Awaiting Recommendations Application Submitted email sent Recommendation request emails sent 8 readers added	may need to check Communications History to see the rec request emails other than the recommenders, it's fine to override other hard fails	Y	
Submit a recommendation through the recommender form	App Status = Awaiting Materials Bin = Awaiting Recommendations Recommendation received emails sent (applicant and referrer versions) still 8 readers	may need to check Communications History to see the referrer version	Y	
Upload a material with type = SIIS - Additional Reference	App Status = Awaiting Decision Bin = Complete still 8 readers	Leave the other checklist item as awaiting	1st attempt: N 2nd attempt: Y	App status stayed at Awaiting Materials; I built a new rule in prod and test that is SIIS-specific and accounts for an uploaded SIIS - Additional Reference material, and also adjusted the general Awaiting Materials rule to exclude SIIS.
Delete one recommendation material (or reset the checklist item)	App Status = Awaiting Materials Bin = Awaiting Recommendations still 8 readers		Y	
Submit/upload the second recommendation again	App Status = Awaiting Decision Bin = Complete still 8 readers		Y	
Add/confirm/release a Withdraw decision	App Status = Decided Bin = Withdraw No readers		Y	

ADMISSIONS APPLICATION PROCESSING & DECISION RELEASE

Scenario	Expected Application Fee	Notes	Pass?	Test Record
Submit an application that is not a PHA app	no fee added		Y	[redacted]
Submit a PHA application where the waiver level is full	no fee added		Y	
Submit a PHA application where the waiver level is 75%	$25 fee added		Y	
Submit a PHA application where the waiver level is 50%	$50 fee added		Y	
Submit a PHA application where fee paid prior cycle is Y	$50 fee added		Y	
Submit a PHA application with no waiver or prior fee paid	$100 fee added		Y	
Submit a PHA application where the waiver level is full and fee paid prior cycle is Y	no fee added		Y	
Submit a PHA application where the waiver level is 75% and fee paid prior cycle is Y	$25 fee added		Y	
Submit a PHA application where the waiver level is 50% and fee paid prior cycle is Y	$50 fee added (comment should reference reduced 50%)		Y	

HOW TO TEST EFFECTIVELY

A great test isn't just about what you test, it's about *where* and *how* you test. Slate gives you multiple environments to work with, and choosing the right one can help prevent data issues, limit disruption, and give you more control over outcomes.

TESTING IN PRODUCTION

Sometimes, testing in production is necessary, especially when a feature can't be accurately simulated elsewhere (like time-based logic or real integrations). If that's the case:

Use clearly labeled dummy records.

Isolate tests with filters or test IDs.

Inform your team to avoid confusion.

Monitor closely for unintended effects.

TEST ENVIRONMENT LIMITATIONS

If there's any risk of affecting live data (especially with forms, rules, scheduled communications, or source formats) it's almost always better to test in a non-production environment. These environments are safer but have restrictions:

No automated imports/exports.

No overnight rule or script processing.

Slower and less reliable rule execution.

Emails do not send.

POST CAPTAIN
CONSULTING

YOUR SLATE JOURNEY *STARTS HERE*

Schedule an intro call with a team that truly understands your work.

LEARN MORE postcaptain.com