PRASHANT BELLAD TEST AUTOMATION CASE STUDY (CYPRESS)

Title

How we helped a Saas product company to empower them to achieve multiple weekly releases through automated E2E test regression that helped *reduce their test regression cycle time by **99%** from **20 hours to 4 mins***

Challenges -

This customer (a product company) wanted to overhaul its product’s UI design, a few user flows and backend services.

Before making these changes live, one of the most important tasks was to have zero impact on existing customers.

As part of this initiative we as a team thus wanted to overhaul the existing E2E test suite so that the team is in a position to proactively

- Catch regression issues
- Execute regression test suite in a quick time
- Make failed tests easy to debug

Goals –

Reduce application regression testing turnaround time and thus help the team to do quick fixes and release with confidence

Solution –

We targeted below weak areas in the current regression testing process which we thought would solve at least 80% of the problem area:

1. Work on application testability
2. Improve the test framework scalability
3. Work on removing test flakiness

PRASHANT BELLAD TEST AUTOMATION CASE STUDY (CYPRESS)

What we did –

- We worked to make the product more testable like implementation of data seeding, stubbing, test data management, one-touch application setup etc
- The first step in test automation was to review and cover the product's critical user flows
- Created E2E Test Automation Strategies and SOPs
- Improved the current Test Framework and made it more scalable. Eg effective use of custom commands etc
- Leveraged the product's testability for automation tests. This helped improve the test automation robustness by at least 60%
- Further analysed and implemented additional effective automation techniques through our QA Continous Improvement Cycle. eg utilising Cypress.io's excellent under-the-hood techniques to improve test automation efficiency & effectiveness
- Enabled the test automation suite with options to run on demand, on PR merge, or at a scheduled time etc as deemed fit
- **Over a period of a few months, 250+ robust E2E tests were added to the suite**
- **We are executing these 250+ tests in 20 parallel threads that get executed in under 4 mins**

Results and Benefits:

✅ 20 hours of regression effort (2.5 PDs) was reduced to 4 mins

✅ We are achieving around $24,000 in annual savings through robust and effective E2E test automation in a team of 2 Test Engineers

✅ Capability developed to get on-demand & quick product health feedback

✅ Continuously improving Product Quality as Test engineers now use the same time to perform break-the-application testing to find edge case issues

✅ You are in a position to make quick decisions

PRASHANT BELLAD TEST AUTOMATION CASE STUDY (CYPRESS)

Sample Test Runs

The below screenshot shows the actual test run results. The breakup of the results:

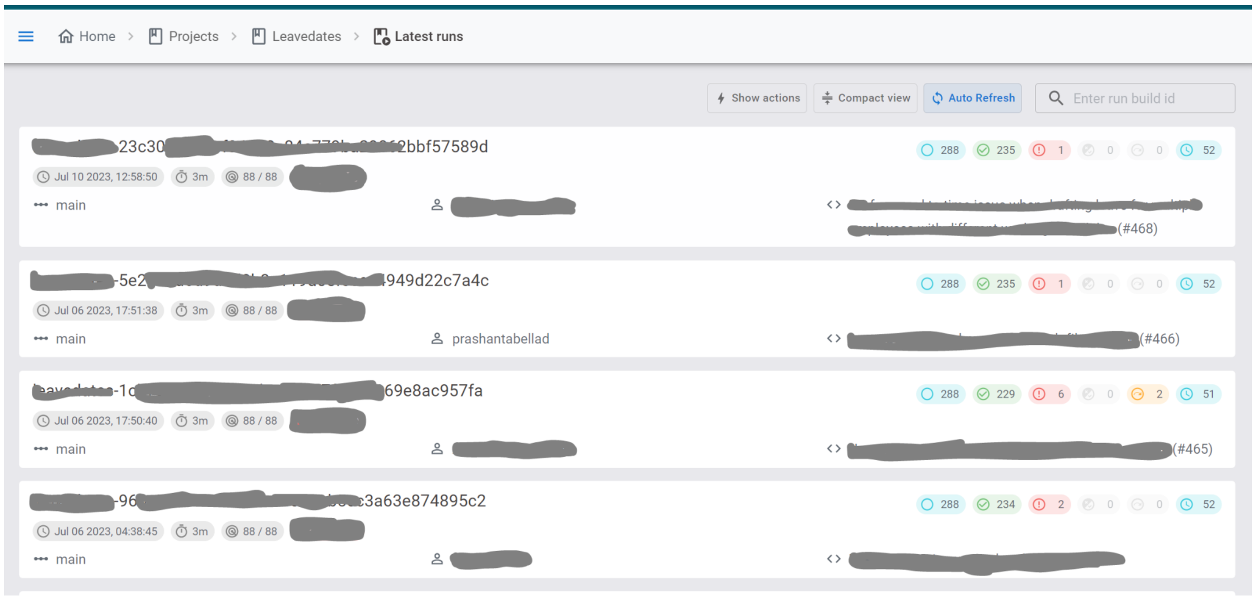
Total Tests: 288

Tests Executed: 236 in under 4 mins

Tests Pass: 235

Failed Tests: 1

Tests TODO (Placeholder for new tests): 52



Home > Projects > Leavedates > Latest runs

Show actions | Compact view | Auto Refresh | Enter run build id

[redacted]23c30[redacted]2bbf57589d — 288 | 235 | 1 | 0 | 0 | 52
Jul 10 2023, 12:58:50 | 3m | 88 / 88 | [redacted]
main | [redacted] | [redacted](#468)

[redacted]-5e2[redacted]949d22c7a4c — 288 | 235 | 1 | 0 | 0 | 52
Jul 06 2023, 17:51:38 | 3m | 88 / 88 | [redacted]
main | prashantabellad | [redacted](#466)

[redacted]-1c[redacted]69e8ac957fa — 288 | 229 | 6 | 0 | 2 | 51
Jul 06 2023, 17:50:40 | 3m | 88 / 88 | [redacted]
main | [redacted] | [redacted](#465)

[redacted]-96[redacted]c3a63e874895c2 — 288 | 234 | 2 | 0 | 0 | 52
Jul 06 2023, 04:38:45 | 3m | 88 / 88 | [redacted]
main | [redacted] | [redacted]

Analysis of Failed Tests in the latest test run

Application intermittent issues (minor): 1

You will observe that for the merge #465 there were 6 failed tests. The tests were quickly analyzed and fixed (the Test team working in parallel with the development team)

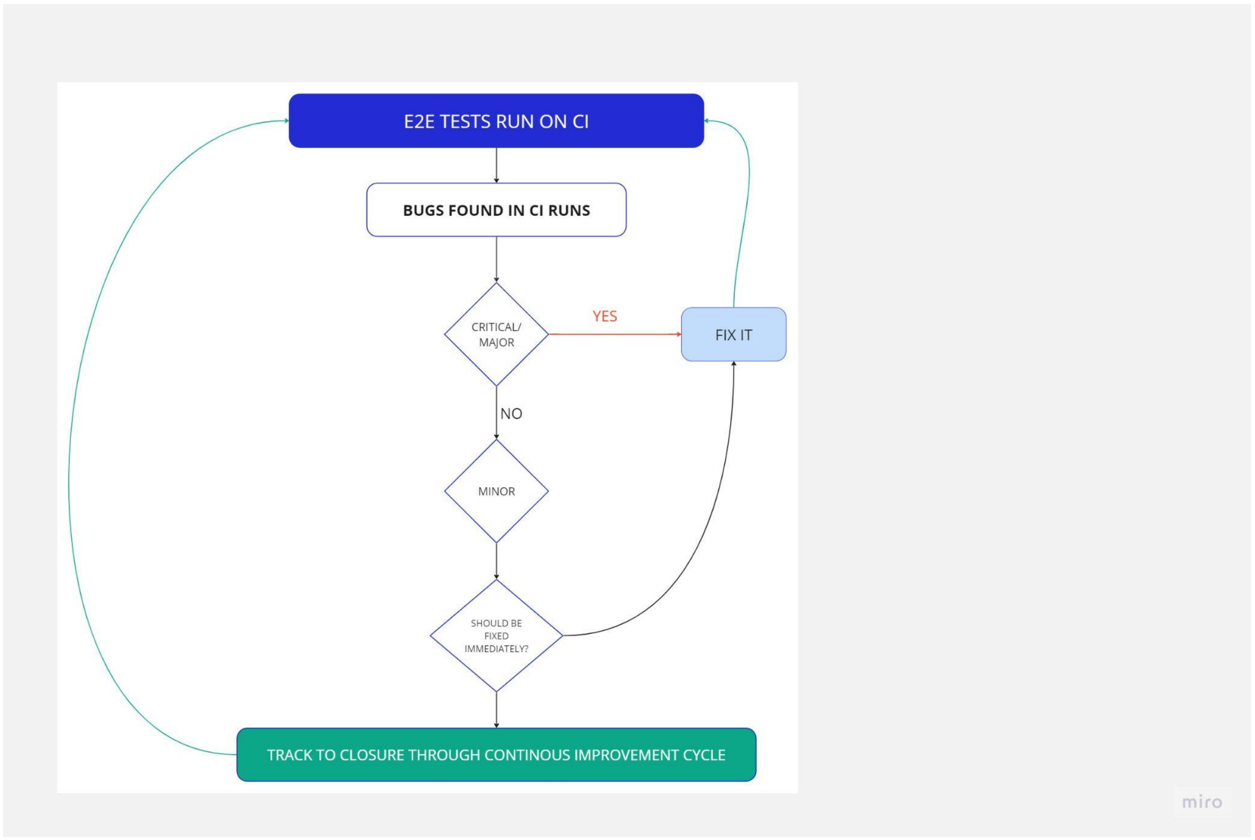
You would be thinking why I am highlighting and analysing failed cases.

PRASHANT BELLAD TEST AUTOMATION CASE STUDY (CYPRESS)

My Strategy for the bugs identified

My approach to automation testing is that it should avoid leaking any bugs to production. If there are any bugs then they should be under control. i.e you have analyzed, classified & acted on them accordingly. Your actions can be fixing them immediately, planning them in a future sprint etc

For this particular company, we decided on the below process:


E2E TESTS RUN ON CI
BUGS FOUND IN CI RUNS
CRITICAL/ MAJOR
YES
FIX IT
NO
MINOR
SHOULD BE FIXED IMMEDIATELY?
TRACK TO CLOSURE THROUGH CONTINOUS IMPROVEMENT CYCLE
miro

PRASHANT BELLAD TEST AUTOMATION CASE STUDY (CYPRESS)

At the same time, there can be two ways to showcase the E2E test's CI run results:

1. Keep highlighting application product issues (if found) in CI results till they are closed

Benefits:

- You **don't lose track of application bugs**, no matter how minor (they are)
- You would **take timely actions** to fix such open issues
- **Avoid accumulation of minor issues**. This helps in achieving a consistent World Class product else your product issues (even minor ones) would eventually lead to a very poor customer experience, **causing a significant dent in your revenues** before you realise it.
- If the Engineers creating automation tests are encouraged to find/ showcase minor functional or technical debt issues then these **Engineers would 100% focus on NOT just PASSING the tests but identify/find issues. This is a great step in becoming a World Class Product!**

Drawbacks:

- Can delay your product releases if your QA strategy is to fix all the minor bugs before a release
- Can block merges if there is a mandatory check for all E2E tests to pass before PR merges. You would need to force merge (manually)

2. Skip the failing tests till they PASS & track them separately to closure

Benefits:

- Passing CI runs would help automate your PR merges
- The minor issues fixes are taken care of by tracking the failing tests separately to closure
- The Test Dashboard looks green and clean

Drawbacks:

- The percentages tell that ignored or skipped issues are not promptly tracked to closure. Most often these issues get forgotten after a few months
- The count of such minor issues (however minor) keeps on growing over a period of time and if not actionized can certainly lead to product failure
- This approach does not present the TRUE picture of your product's health. This may not be a concern when the product and its customers are tiny, but it will certainly hurt badly as the product and its customers grow in size.