

REPORT PORTAL POC

Jul 2023



Paramount

CBS

SHOWTIME



nickelodeon

MV

COMEDY
CENTRAL

BET



plutoTV

Simon &
Schuster

5

10

telefe

Chile
Visión

colors

OBJECTIVE

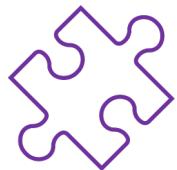
1

IMPROVE INSIGHTS
ON AUTOMATION
FAILURES ON THE
NEWRELIC
DASHBOARD



2

EFFICIENTLY HANDLE
AUTOMATION FAILURES
TO MAINTAIN AND
IMPROVE THE PASS RATE



ABOUT REPORT PORTAL

AI-powered Test Automation Dashboard

acquire, aggregate and analyze test reports
to ascertain release health

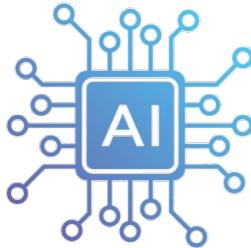


Key Highlights

- **Free & Open source** (AWS infra cost to host will be applicable)
- **AI/Machine learning** based failures analysis (kNN + Gradient Boosting)
- **Centralized reporting** and Better management
- Better **QE collaboration** for tackling of failures & script maintenance work.
- **Better insight** on Bugs, Automation issues, and System issues
- **API interface**, which can be used to pull data (with analysis/failure classification info) and push to NewRelic for centralized dashboard.
- **Historical data** of test execution and analysis, can be retained forever.
- **~50% or more effort reduction for failure analysis**. Delegate a portion of analytics work to artificial intelligence.
- **Reduce time** and cost for manual analysis through **intuitive UI** for failure analysis
- **Real Time Results** on ongoing executions.
- **Widgets and dashboards** that can help with automation maintenance.



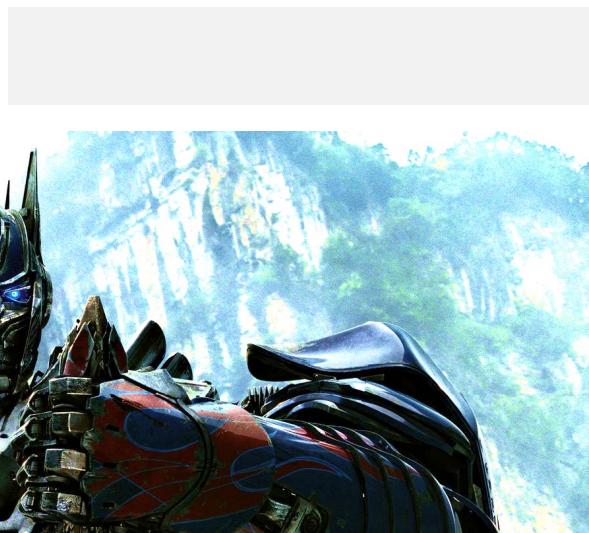
PRE-REQUISITE



GARBAGE IN, GARBAGE OUT



*Transform test script
to be better, efficient
and accurate on
catching and
reporting
issues.*



1

APPROPRIATE
STEPS AND LOG
STATEMENTS IN
THE TEST FOR
CLEAN STEP ON
REPORTING TOOLS.



2

AVAILABILITY OF ALL
THE PROPER OR
REQUIRED
VALIDATION IN TESTS,
AND ACCURACY OF
FAILURES AND
RELATED MESSAGE

Appropriate and accurate message on assert statements

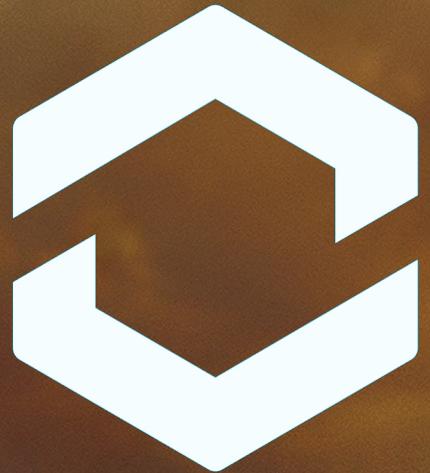
```
softAssert.assertTrue(homeScreen().isErrorIconPresent(), message: "Error page icon is not present.");
softAssert.assertTrue(homeScreen().isErrorButtonPresent(), message: "Error page ok button is not present.");
softAssert.assertTrue(homeScreen().isErrorTitlePresent(), message: "Error page error message is not present.");
```

Appropriate method segregation and correct step logging

```
1 usage  ~ sreegop +5*
@Step("Verify invalid password message")
private void verifyInvalidPasswordMessage(SoftAssert softAssert) {
```

Appropriate checks in place during navigation to fail with correct issue

```
mainMenuNavigation().openItem(MenuItem.SERIES);
verifyPageLoad(seriesScreen());
validator().validateEdenParameters(EdenEvent.NAVIGATION_OPEN.getType(), getExpectedEdenNavigationSeriesPageOpen());
```



reportportal.io
**QE AUTOMATION
ENPOWERMENT**



FAILURE INSIGHTS

Detailed insights on failures /
Customizable sub classification options
based on project needs

- Options are customizable based on project needs
- Allows up-to 14 subtypes

↓ This is on scrum team

DEFECT NAME	ABBREVIATION	COLOR	DIAGRAM
PRODUCT BUGS GROUP			
Product Bug	PB	●	
+ Add Defect Type	14 subtypes can be added		
SYSTEM ISSUES GROUP			
System Issue	SI	●	
Synergy Issue	SY	●	
Framework Issue	FI	●	
Service Issue	SE	●	
+ Add Defect Type	11 subtypes can be added		

↑ This is on support team (Services team, Devops, Unified, Synergy)

↓ This is on Automation Engineer

DEFECT NAME	ABBREVIATION	COLOR	DIAGRAM
AUTOMATION BUGS GROUP			
Automation Bug	AB	●	
UI Changes	UI	●	
Missing Data	MD	●	
Session TimeOut	ST	●	
Driver Issue	DI	●	
ReRun Pass	RP	●	
Config Issue	CI	●	
Scripting Error	SE	●	
+ Add Defect Type	7 subtypes can be added		



AUTO ANALYSIS

Failure Auto analysis (AI based and having accuracy) and auto tagging and effort saving on large

- Quick analysis of approx. 50% - 60% failures in a large build, which can improve over time.
- Regular pipeline can rarely have something new to look at

NAME	START ▾	TTL	PS	FL	SKP	PRODUCT BUG	AUTO BUG	SYSTEM ISSUE	TO INVEST	□
Unified_ANDROID_TV_MTV #48	10 days ago	155	95	57	3					<input type="checkbox"/>
🕒 6h 55m	📍	👤 default								

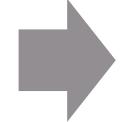
AI/ML Auto analysis Implementation Details: <https://www.youtube.com/watch?v=2eXKqHOPgWA>



INTUITIVE ANALYSIS INTERFACE

For manual analysis, report portal comes with intuitive UI with similar failure information, and auto analysis match details which did not match 80% threshold

Potential failure reasons from ML, for analysis support / easy tagging which did not meet the threshold config 80% for auto tagging.



Quick analysis tagging of all failures on the build with failure on same step.



The screenshot displays two side-by-side panels of a software interface:

- Execution to change:** This panel shows a test step named "cwStartsFromSavedPositionTest" with a status of "To Investigate". Below the step name, there is a truncated error log:

```
com.viacom.unified.exceptions.MQETestException: Video player wasn't
opened.
at
com.viacom.unified.ui.unifiedscreens.PlayerScreen.waitForVideoLoaded(P.....
```

A dropdown menu titled "Apply for: Similar 'To Investigate' in the launch & current item" lists "longMovieNameAndDescriptionTest" as an option. Two checkboxes are present: "0/1 items selected" and "Show Error Logs".
- Select defect:** This panel features a "Manual SELECTION" button highlighted in teal. It includes a "77% ANALYZER SUGGESTION" card and a "76% ANALYZER SUGGESTION" card. A section titled "Select defect manually" contains a grid of 15 colored circles, each representing a defect type: Product Bug (orange), Automation Bug (yellow), System Issue (blue), No Defect (grey), UI Changes (light green), Synergy Issue (medium green), Missing Data (orange), Framework Issue (purple), Session TimeOut (brown), Service Issue (teal), Driver Issue (pink), ReRun Pass (light blue), Config Issue (dark purple), and Scripting Error (light orange). A checkbox labeled "Ignore in Auto Analysis" is located on the right, and a "To Investigate" button is highlighted in teal at the bottom right.

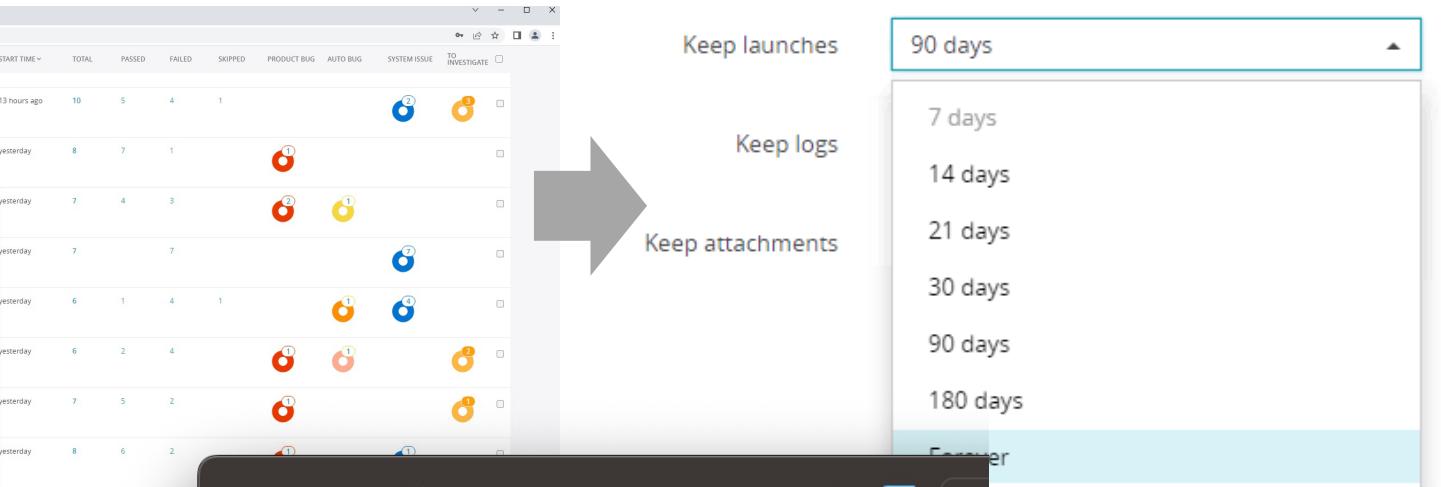
TEST RUN CENTRAL REPOSITORY

Central repository
and lifespan/history
as we wish vs 10-
day allure report.

The screenshot shows the Synergy reporting interface. On the left, there's a sidebar with navigation links like Overview, Categories, Graphs, Timeline, Behaviors, and Packages. The main area displays 'PARAMOUNT TEST RESULTS 7/13/2023' with 15 test cases (80% green). It includes a donut chart, a 'TREND' chart showing a recent increase in defects, and sections for 'SUITES', 'ENVIRONMENT', and 'EXECUTORS'. The 'ENVIRONMENT' section shows appPackage and appVersion details. Below this is a detailed test run log with steps like 'Launching the app' and 'Initial Navigation to Home Screen', each with a status of 'PASSED' and execution times.

- Centralized portal for all runs vs independent reports links for each run.

- Flexible retention store for certain period or forever based on the storage available.



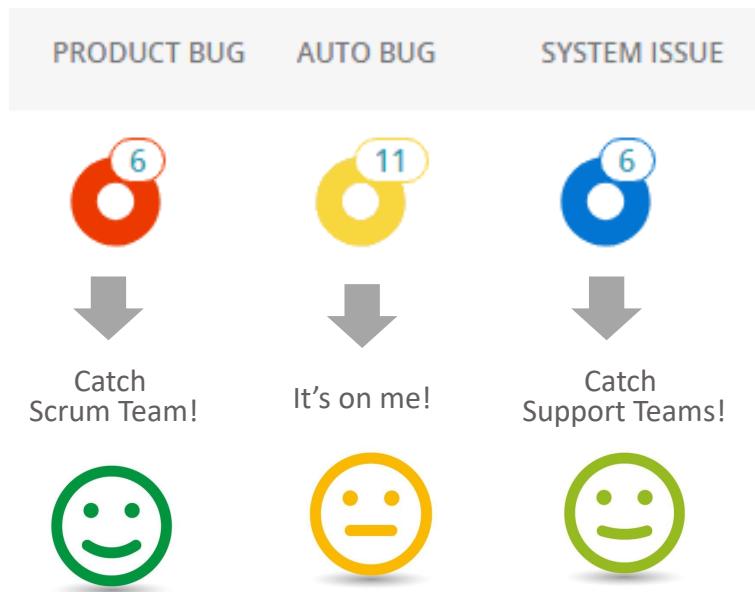
- Avoided media file attachment, and linked to synergy for optimized storage / bandwidth usage.



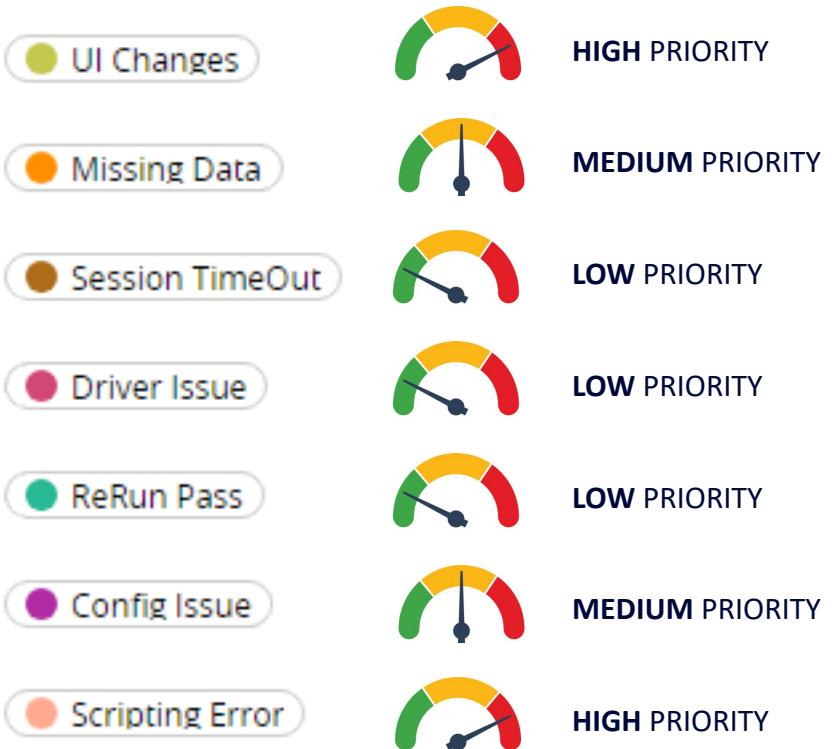
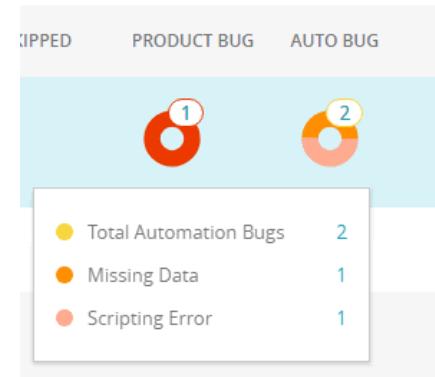
FAILURE TRACKING

Better tracking and prioritization of automation failures and categories. (Automation maintenance, follow-up on bug fixes, Synergy issues follow-up etc).

- With the detailed insights on the failures, addressing them got even more easier.



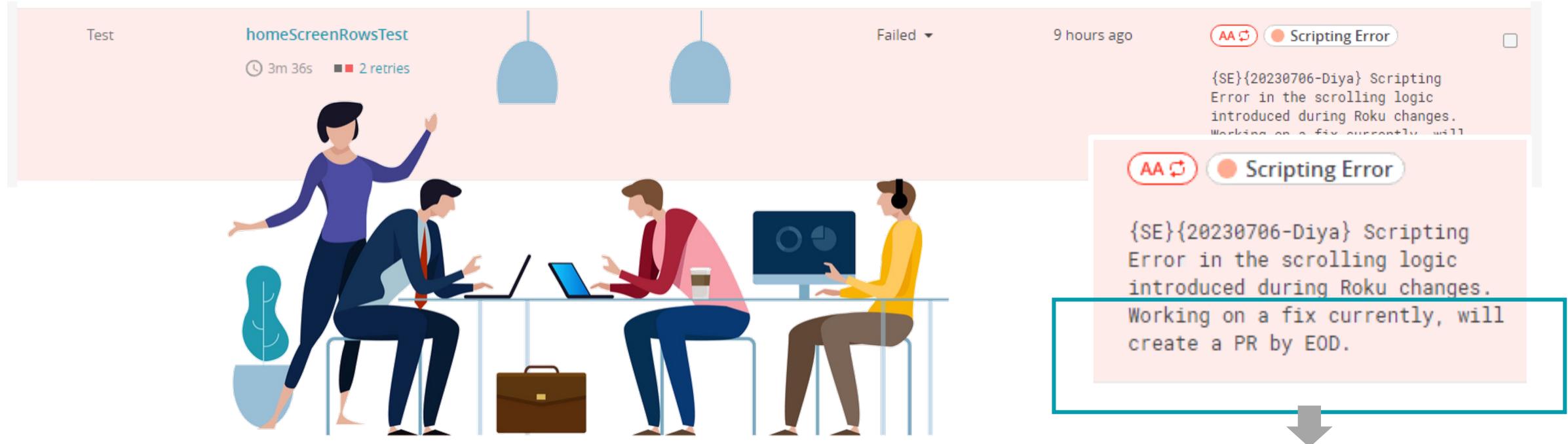
- Can achieve efficient planning on maintenance with detailed insights on failures



QE COLLABORATION

Team can even collaborate on maintenance by adding appropriate comments on fix progress on latest report portal build.

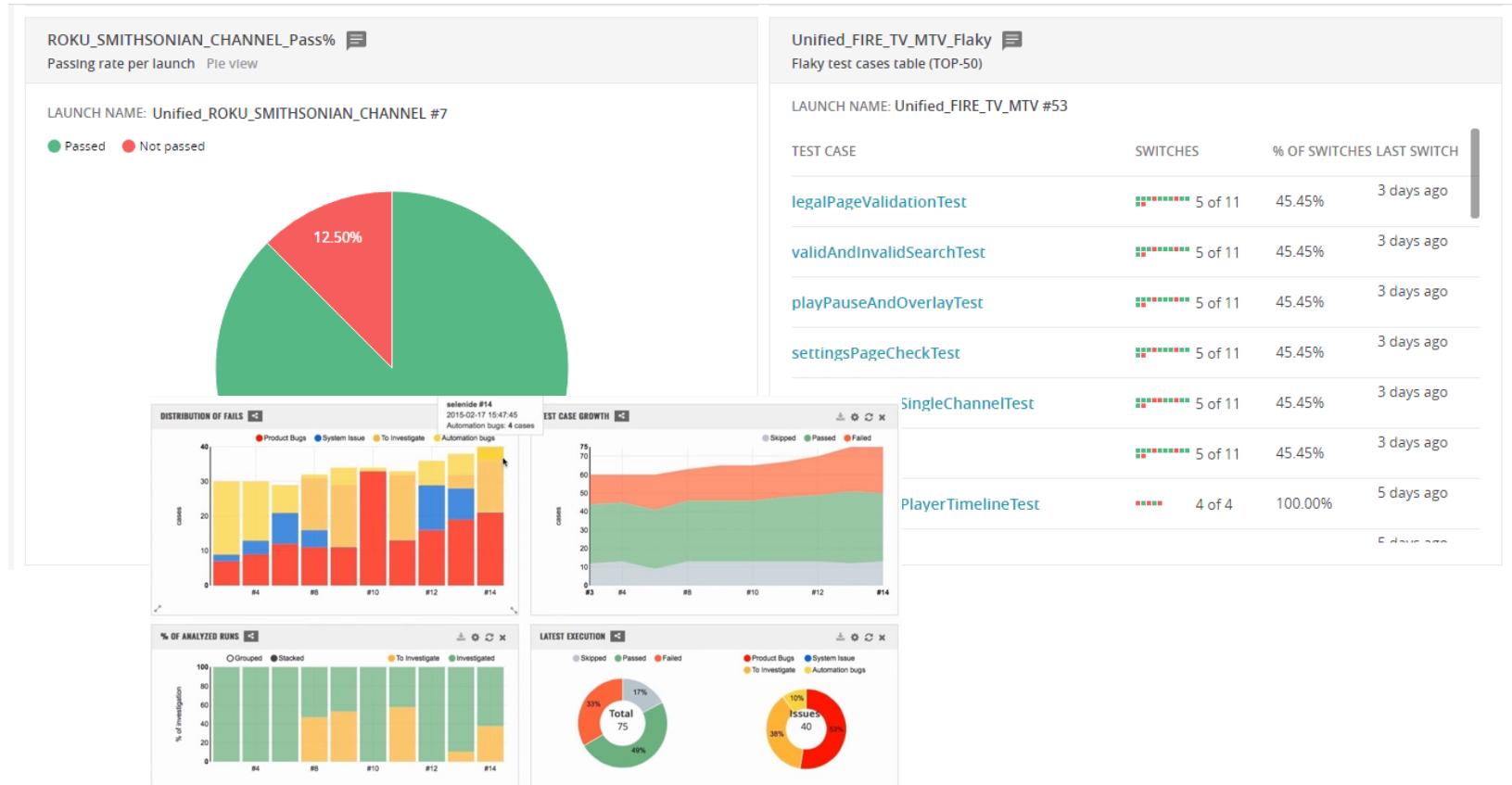
- improve maintenance turnaround and avoid conflict in maintenance fixes



- Fix Status / Progress by other team members.
- Do not pick this for fixing.

AUTOMATION DASHBOARDS

Dashboards that can help with automation insights and maintenance progress to effectively handle backlog.



Note:

- These are focused on automation stability and quality to help with maintenance
- New Relic Dashboards will be providing insights on Application quality and overall QA statistics from various sources.



QE DASHBOARD INTEGRATION

API support to extract data and push to New Relic to enhance QE Dashboards.

API DOCUMENTATION

DL

Activity Controller

Bug Tracking System Controller

Dashboard Controller

Demo Data Controller

File Storage Controller

Integration Controller

Report Portal 5.8.0
[Base URL: 10.242.107.234:8080/api]

Report Portal API documentation

Contact Support

Apache 2.0

activity-controller : Activity Controller

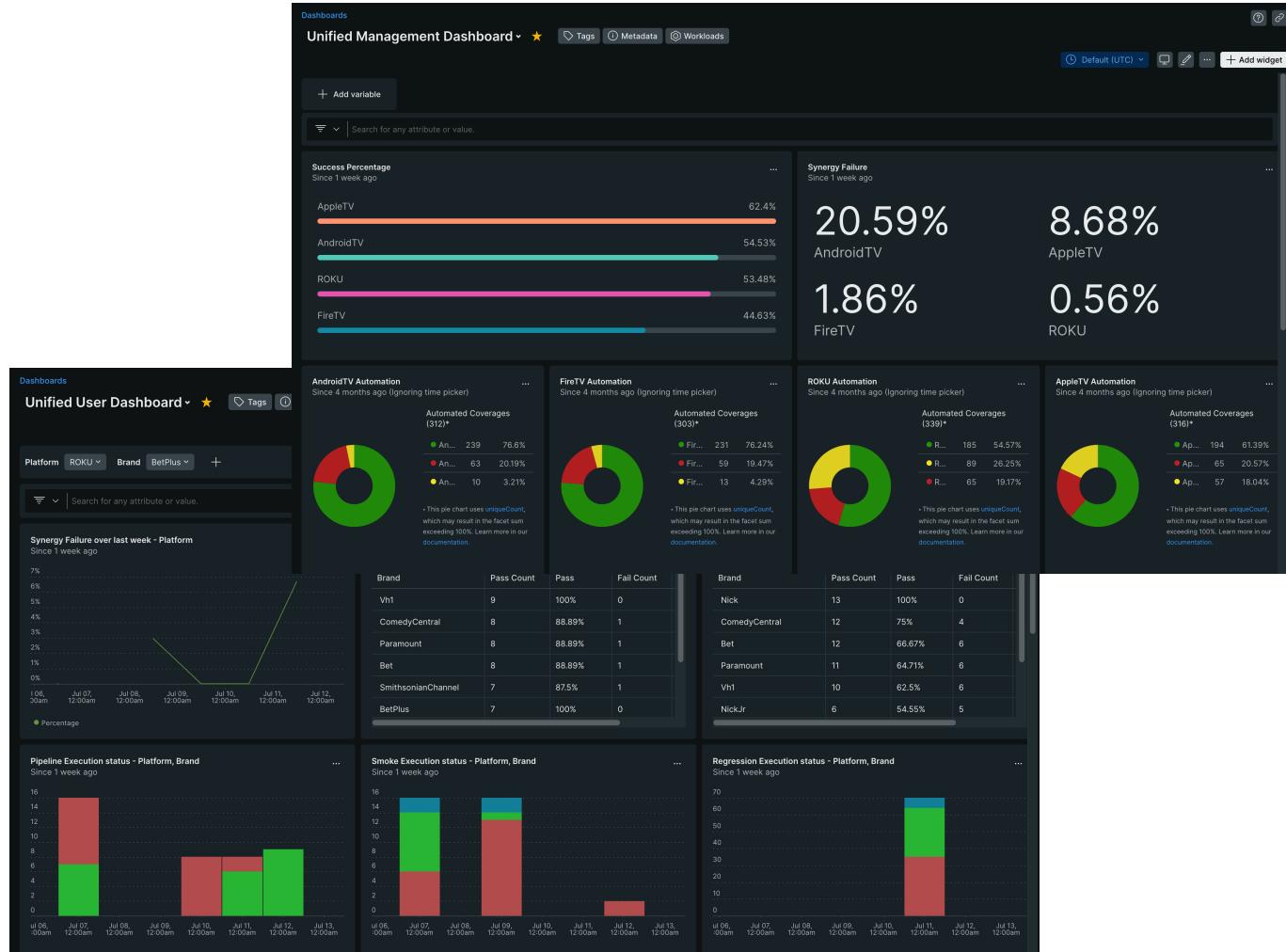
bug-tracking-system-controller : Bug Tracking System Controller

dashboard-controller : Dashboard Controller

demo-data-controller : Demo Data Controller

file-storage-controller : File Storage Controller

integration-controller : Integration Controller





BEST PRACTICES



CODE FOR CODING

Merged

UO-758 Report portal Step Addition #1020

rob1nalex merged 3 commits into master from UO-758 3 days ago

```
// Modify the content
if (content.toString().contains("@Step")) {
    fileCount++;
    System.out.println(fileCount + ". " + file.getAbsolutePath());
    String modifiedContent = content.toString().replace("import io.qameta.allure.Step;\n", "");
    String[] modifiedContentArr = modifiedContent.split("\n");
    ArrayList<String> finalContentList = new ArrayList<>();
    for (int i = 0; i < modifiedContentArr.length; i++) {
        if (modifiedContentArr[i].contains("@Step()")) {
            String strMsg = modifiedContentArr[i].replace(")", ";");
            String methodName = "";
            for (int j = i; j < modifiedContentArr.length; j++) {
                if (modifiedContentArr[j].contains(") {")) {
                    String methodLine = modifiedContentArr[j];
                    methodName = methodLine.split("\\(")[0];
                    methodName = methodName.substring(methodName.lastIndexOf(" ") + 1);
                    break;
                }
            }
            String literalName = convertToLiteralName(methodName) + "_STEP";
            String lineIndent = strMsg.split("@")[0];
            strMsg = strMsg.replace("@Step(", "private static final String " + literalName + " = "
            finalContentList.add(strMsg);
            finalContentList.add(lineIndent);
            finalContentList.add(lineIndent + "@io.qameta.allure.");
            finalContentList.add(lineIndent + "@com.epam.reportport");
        } else {
            finalContentList.add(modifiedContentArr[i]);
        }
    }
    modifiedContent = String.join("\n", finalContentList) + "\n"
    // Write the modified content back to the file
    BufferedWriter writer = new BufferedWriter(new FileWriter(file));
    writer.write(modifiedContent);
    writer.close();
}
```

294	311	
295	-	@Step("Enter new password '{password}'")
312	+	private static final String ENTER_NEW_PASSWORD_STEP = "Enter new password '{password}'";
313	+	
314	+	@io.qameta.allure.Step(ENTER_NEW_PASSWORD_STEP)
315	+	@com.epam.reportportal.annotations.Step(ENTER_NEW_PASSWORD_STEP)
296	316	public void enterNewPassword(String password) {
297	317	if (TestRunInfo.isRoku() && newPasswordField().isPresent()) {
298	318	enterPassword(password);

Created Java program to scan the code in repository and modify to report to both report portal and allure Step annotations.



Code updated by code

+2,963 -898 █████

Changes within a day

<https://github.com/viacomcbs/mqe-unified-oao-tests/pull/1020/>



QE GUIDELINE



Created QE guideline to get best out of tool in supporting QE automation activities and achieve best quality for application and automation scripts.

Report Portal Test Status Update Guidelines

File Edit View Insert Format Tools Extensions Help

100% Arial 14 + B I U A

Test Status Update Guidelines

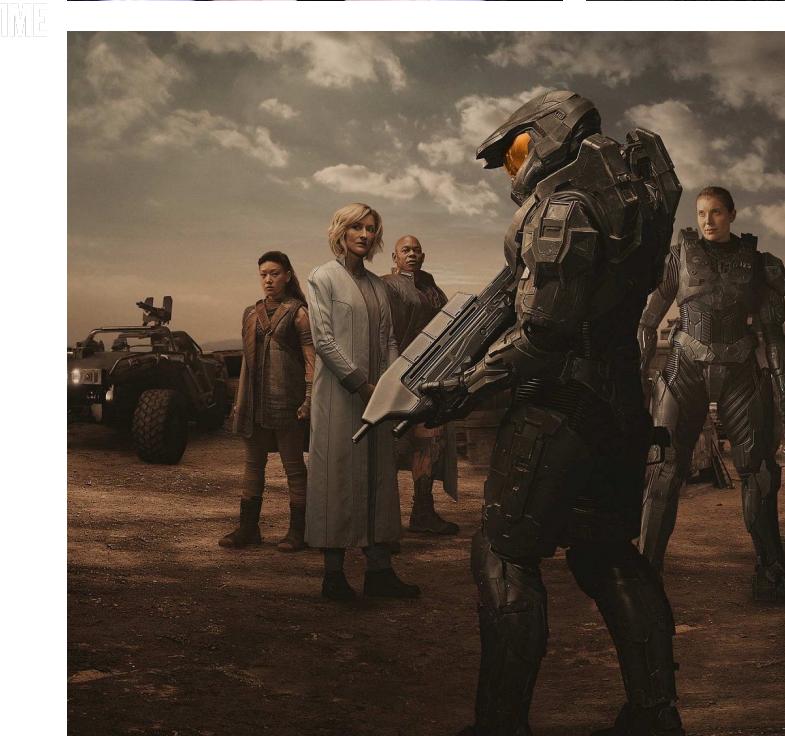
General Guidelines

- Review items tagged as To Investigate first. Once you are done with To Investigate items, review Product bugs, make sure all the defects are logged and open (if not, take appropriate actions). After that, review remaining Auto-Analyzed items tagged by system, and see if Analysis still holds good. After your review, cleanup any discrepancies in the comments and make sure it's up to date.
- Tagging should be based on the cause of failure for a specific run. If it fails on another step during rerun after fix; keep the tagging based on the root cause of failure, and add details on further failures as comments. This is to keep the Auto tagging by system as accurate as possible for future failures.
- If you can't conclude on the root cause of failure, which requires detailed analysis or clarification. Keep it in To Investigate with details added in comments for investigation done.
- No need to keep the history log in the latest comment. Merge all relevant info into a single comment, which is relevant and latest. History is tracked by system and accessible; keep the latest comments clean.
- Follow the format: {#RP Tag}{YYYYMMDD-firstname.lastname} what caused the failure? What action needs to be done to fix the issue (not required for product bug) code fix / Rerun details.

Guidelines on Tagging

- Product Bug (only one tag level)
 - This tag is used when the script has failed due to a product bug.
 - Make sure you are cross checking Auto tagged Product bugs and revisiting the

https://docs.google.com/document/d/16_UmN9vdwUblvF6vqe_of4P5skku9WliRxT9h1QSbY/



THANK YOU

Paramount

