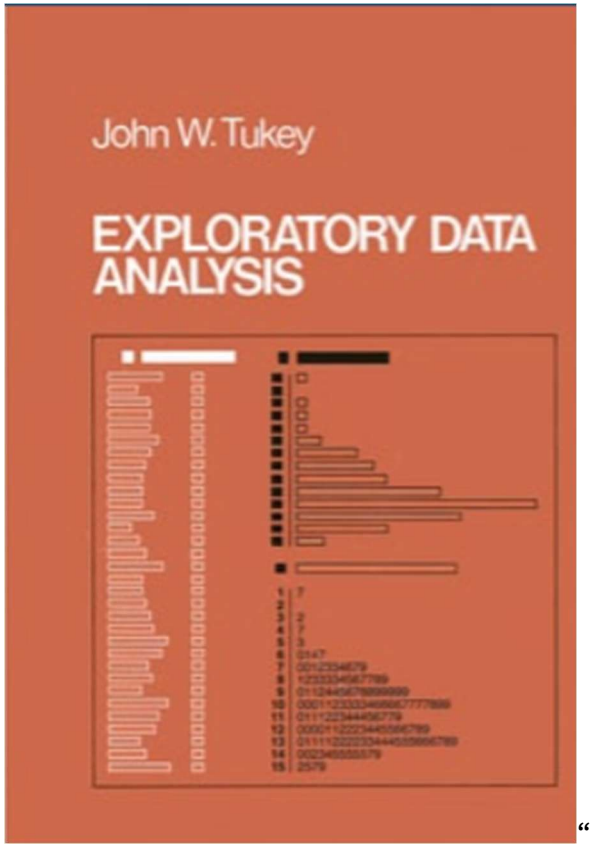
EDA for QC of ADaM

Trust requires traceability

James Joseph

- EDA for QC of ADaM

EDA



“Don’t expect standard summaries to reveal the unusual.”

QC Levels

- L1 = self-review
- L2 = peer-reviewed
- L3 = double-coded, compared

*QC level is not enforced, only planned

The Problem We Solve

“We're finding same issues, again?!”

- QC logs scattered, usually Excel
- Task tracking is inconsistent
- Verification is hard to trace
- Issues repeat across cycles

What We've Learned

Actually Quadruple Programming!

- Excel doesn't scale
- Traceability is non-negotiable
- Double-Independent QC = Biased.
- Roles must be enforced

EDA Solution
Event-driven architecture is a software design model built around the publication, capture, processing and storage of events.

Where EDA Fits In

Extract → **Transform**
Between raw input and formal output.
→ Load

Where EDA Fits In

QC Plan → QC Log + Deliverables
→ Live Tracker → **Archive**

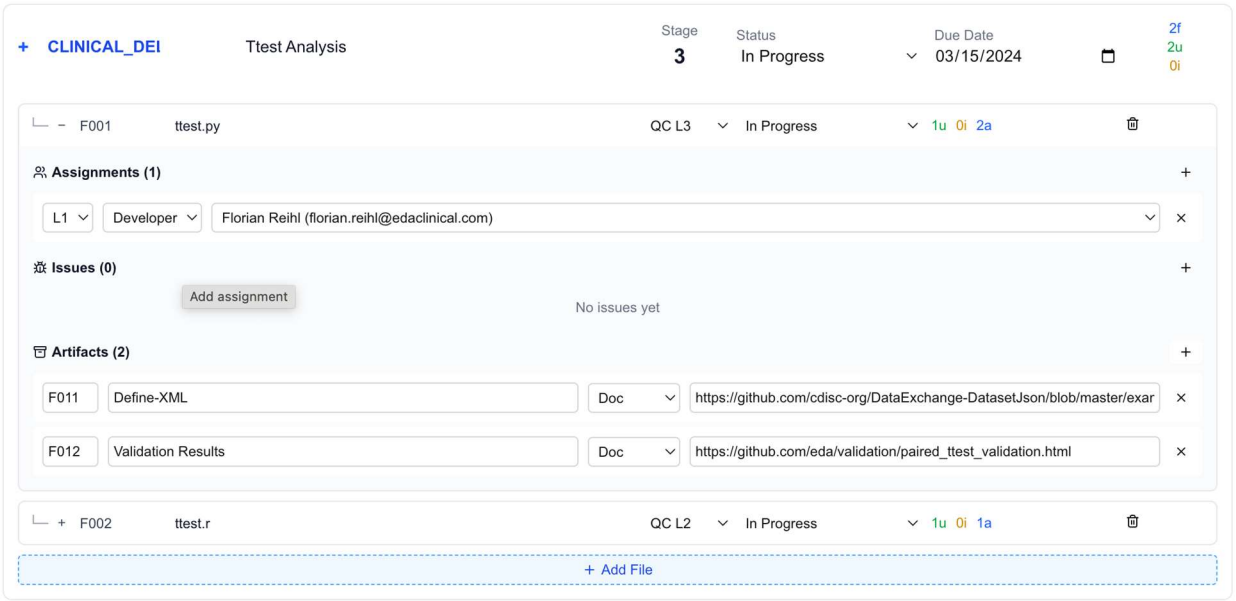
Who This EDA For

Reviewers

- Biostatisticians verifying deliverables
- Programming managers
- QC leads and reviewers

Our Terminology

- **Study** = container
- **QC Plan** = qc levels + assignments
- **Deliverable** = grouped files
- **File** = assigned unit
- **Artifact** = required file



Enforcing QC Level

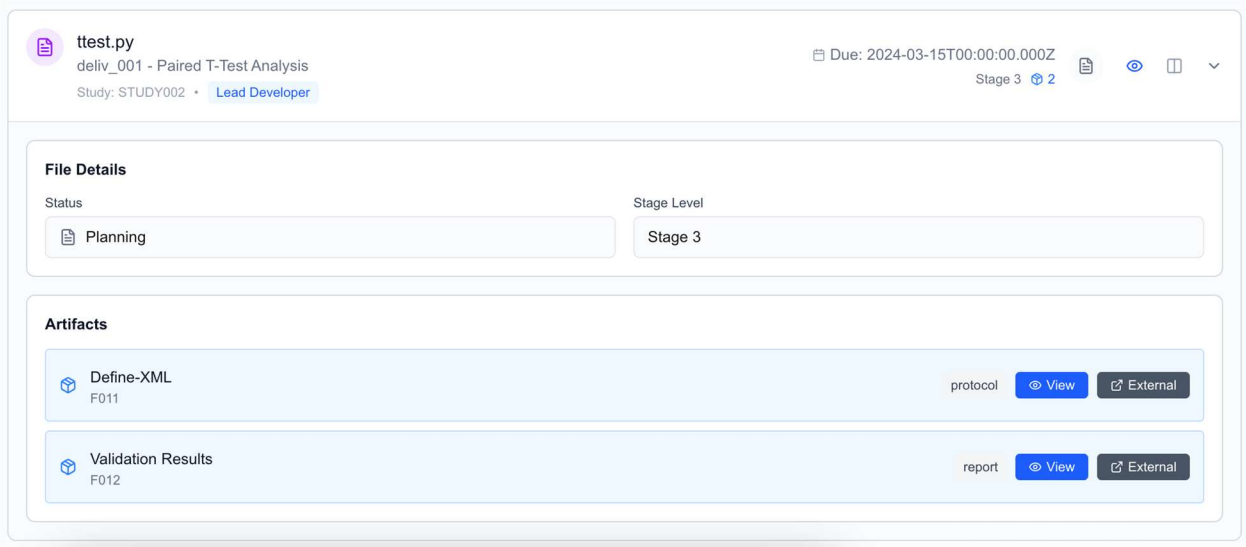
- **L1** = self-review•,
- **L2** = peer-reviewed
- **L3** = double-coded, compared

NOTE: not optional;

Our Roles

- **Manager**: assign, compare, approve
- **Developer**: implement
- **Lead** = Developer + Manager
- **Reviewer**: approve
- **Admin**: assign roles in system

Developer View of Tracker



How QC Levels Are Enforced

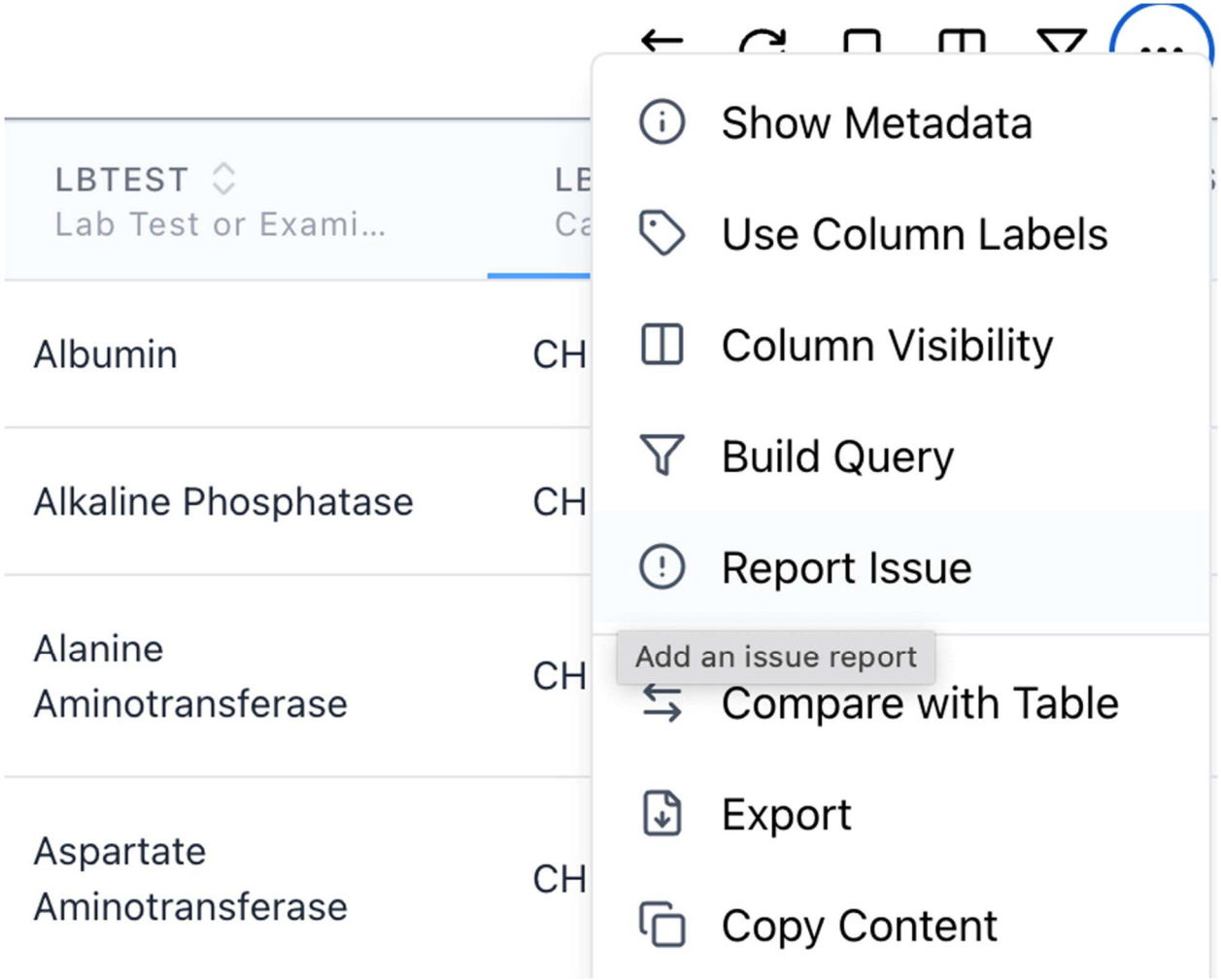
Role boundaries protect Level 3 QC integrity:

- Developers can’t view other developers’ files.
- Developers cannot run compares on their own outputs.
- Developers do not share the same input files.

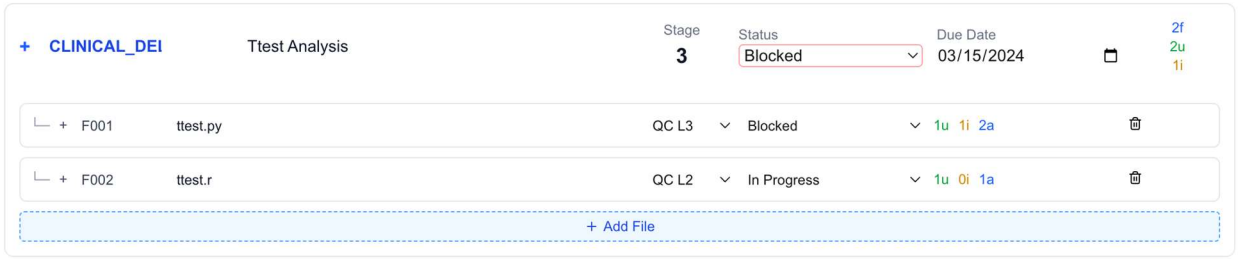
Triggers

- File assigned by manager, e.g. upload/update qc plan →
- Event = logged activity

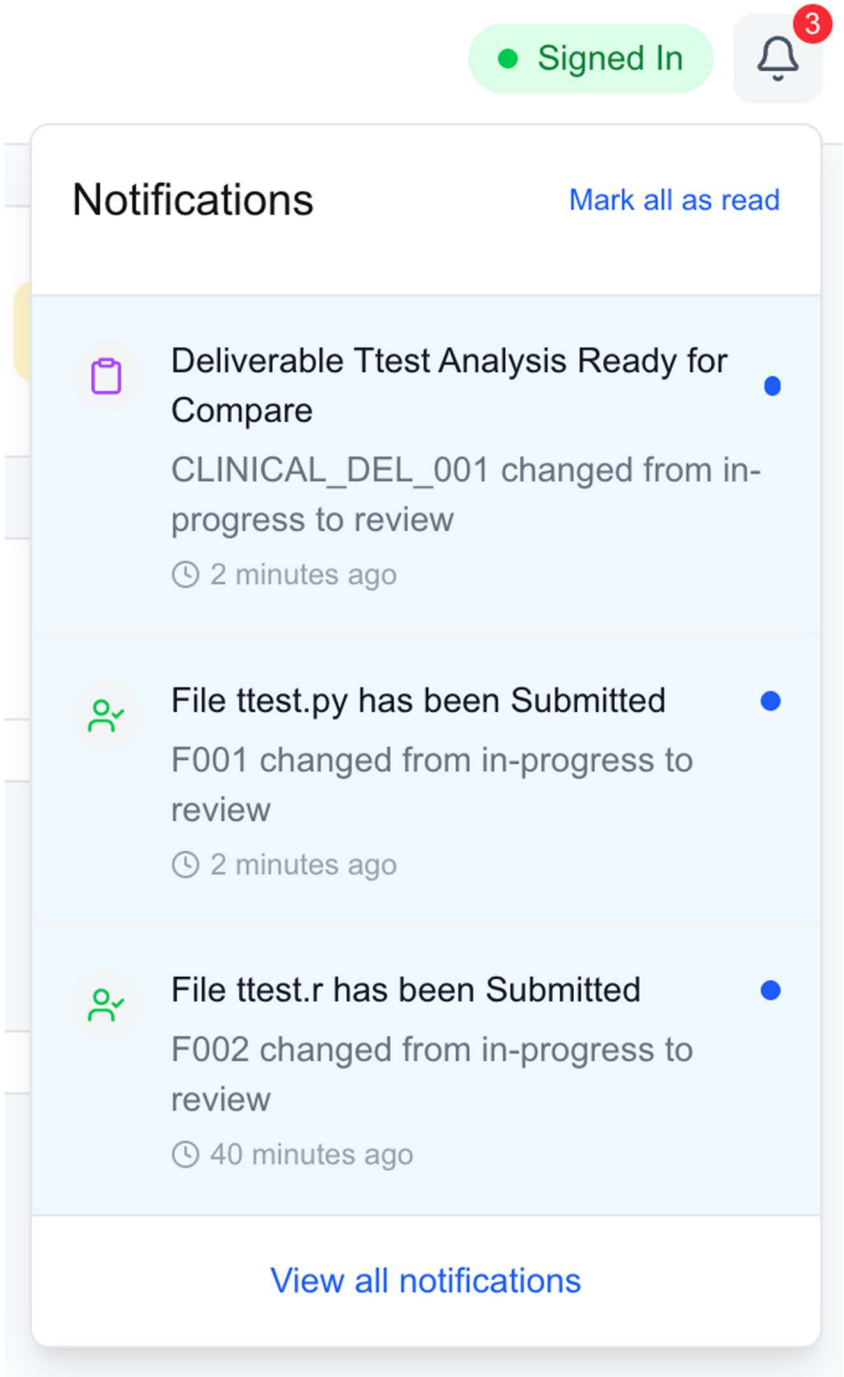
IssueLogged



StatusChanged



Notification of Event


Signed In
3
Notifications
Mark all as read
Deliverable Ttest Analysis Ready for Compare
CLINICAL_DEL_001 changed from in-progress to review
2 minutes ago
File ttest.py has been Submitted
F001 changed from in-progress to review
2 minutes ago
File ttest.r has been Submitted
F002 changed from in-progress to review
40 minutes ago
View all notifications

Tracker
QC Plan in Action

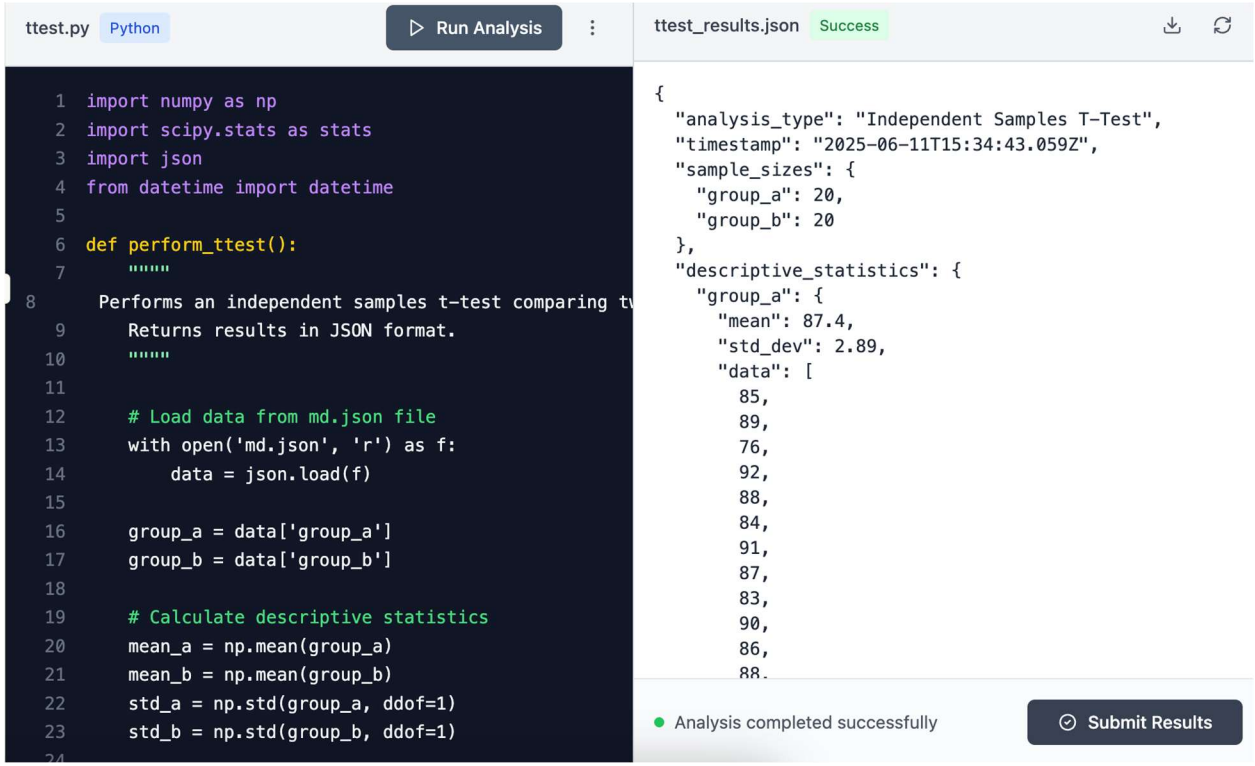
- NDJSON = full event log
- Reconciler builds tracker from log
- Every change is derived
- No manual edits allowed

Types of events
Setup & Submission Events
Issue Management Events
System-Derived Events

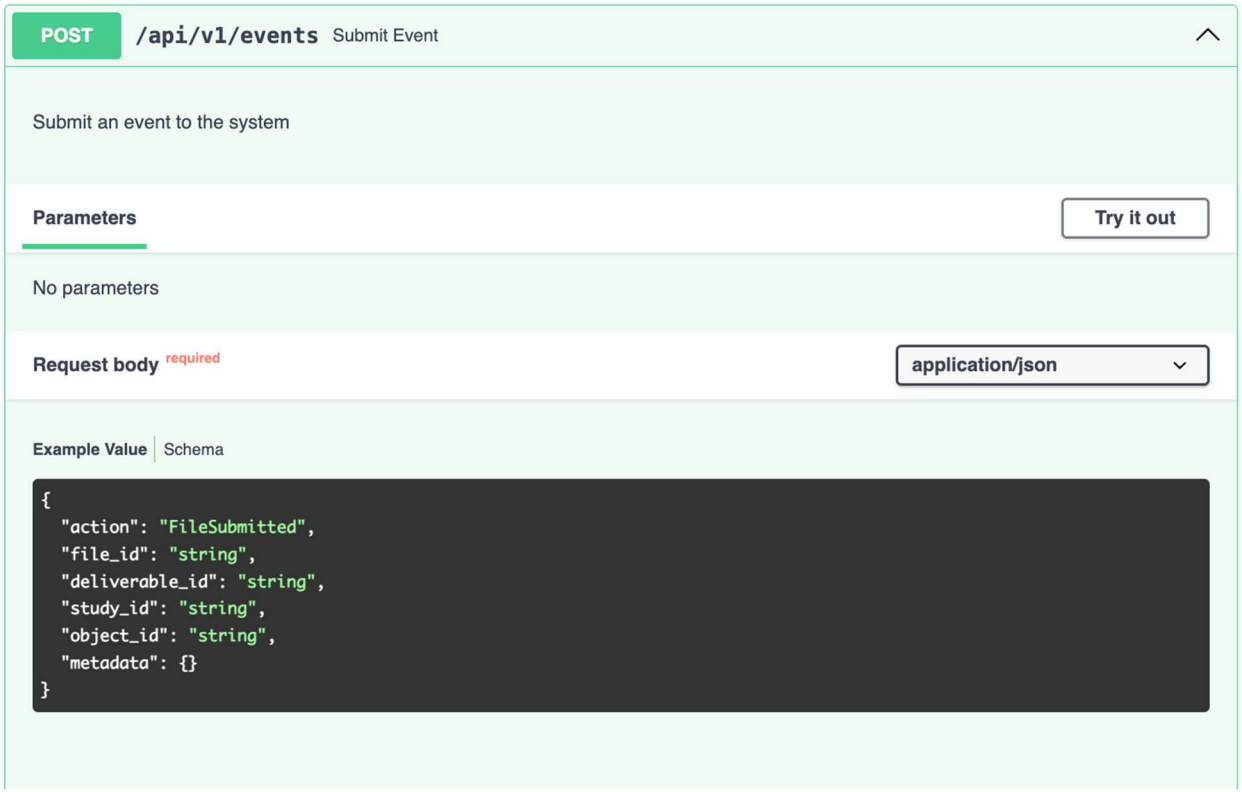
Setup & Submission Events

Event	Purpose	Tracker	Notify?
QCPlanUploaded	Start plan	File view shown	No
FileSubmitted	Submit code or compare	Artifact marked	Yes
DevCommented	Add dev note	Note shown	No

FileSubmitted



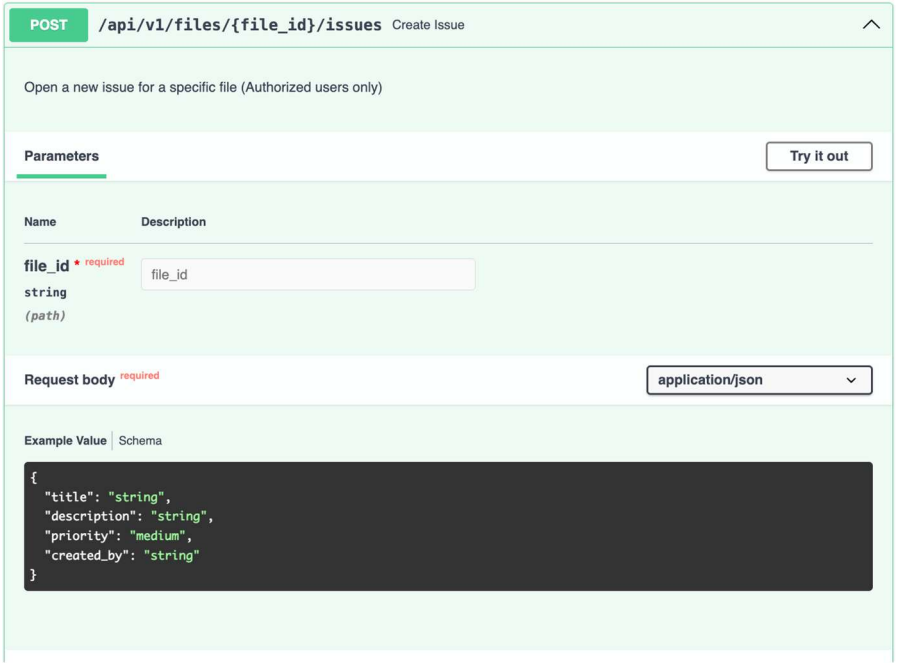
FileSubmitted



Issue Management Events

Event	Purpose	Tracker	Notify?
IssueLogged	Block a file	File = blocked	Yes
IssueResolved	Reopen workflow	File status reevaluated	Yes

IssueLogged



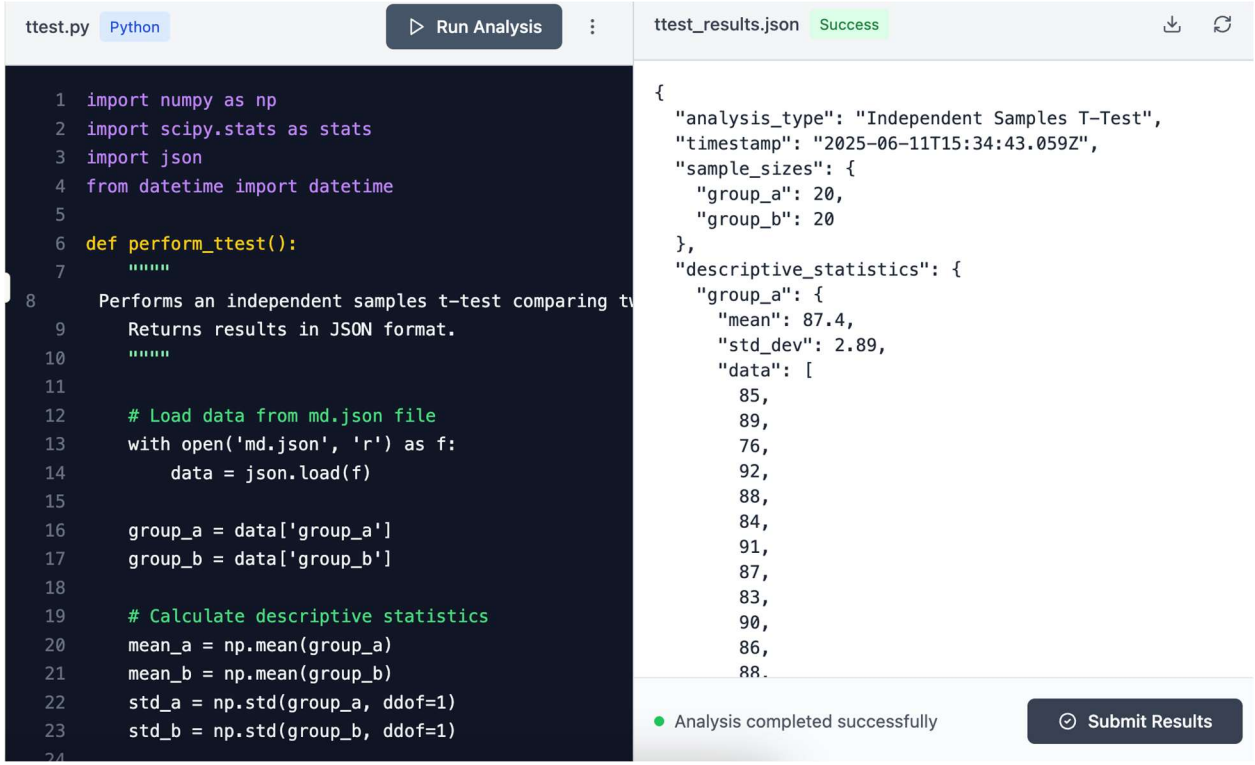
System-Derived Events

Event	Trigger	Tracker	Notify?
StatusChanged	Artifact state check	Status changes	Yes
StageCompleted	All stage files complete	Status changes	Yes
DeliverableComplete	All stages passed	Status changes	Yes

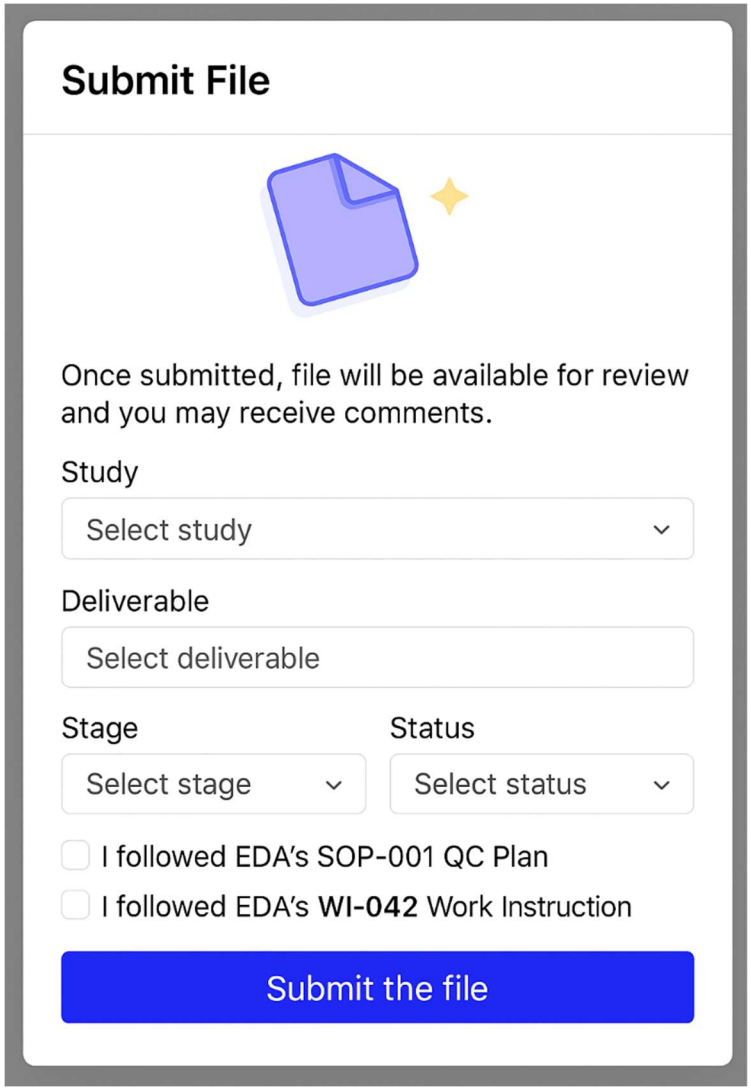
UAT
The CAMIS Project

- Assigned devs to build R + Py tests
- Same input, independent code
- Compare after both uploaded
- Compare result logs QC pass/fail

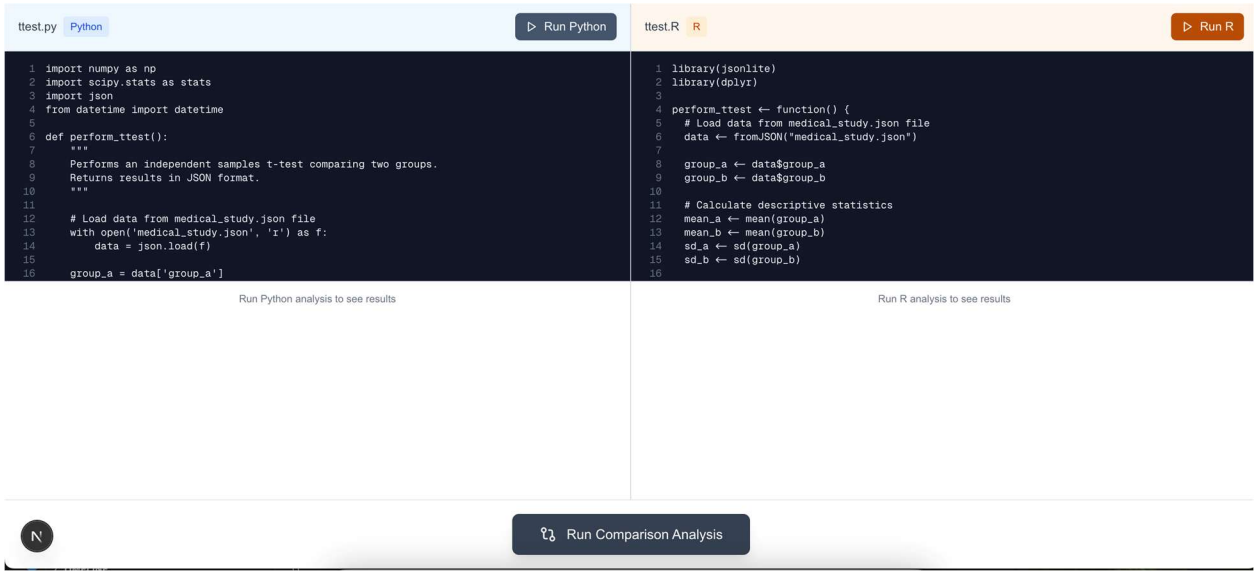
Execute


ttest.py
Python
Run Analysis
ttest_results.json
Success
import numpy as np
import scipy.stats as stats
import json
from datetime import datetime
def perform_ttest():
"""
Performs an independent samples t-test comparing t
Returns results in JSON format.
"""
Load data from md.json file
with open('md.json', 'r') as f:
data = json.load(f)
group_a = data['group_a']
group_b = data['group_b']
# Calculate descriptive statistics
mean_a = np.mean(group_a)
mean_b = np.mean(group_b)
std_a = np.std(group_a, ddof=1)
std_b = np.std(group_b, ddof=1)
{
"analysis_type": "Independent Samples T-Test",
"timestamp": "2025-06-11T15:34:43.059Z",
"sample_sizes": {
"group_a": 20,
"group_b": 20
},
"descriptive_statistics": {
"group_a": {
"mean": 87.4,
"std_dev": 2.89,
"data": [
85,
89,
76,
92,
88,
84,
91,
87,
83,
90,
86,
Analysis completed successfully
Submit Results

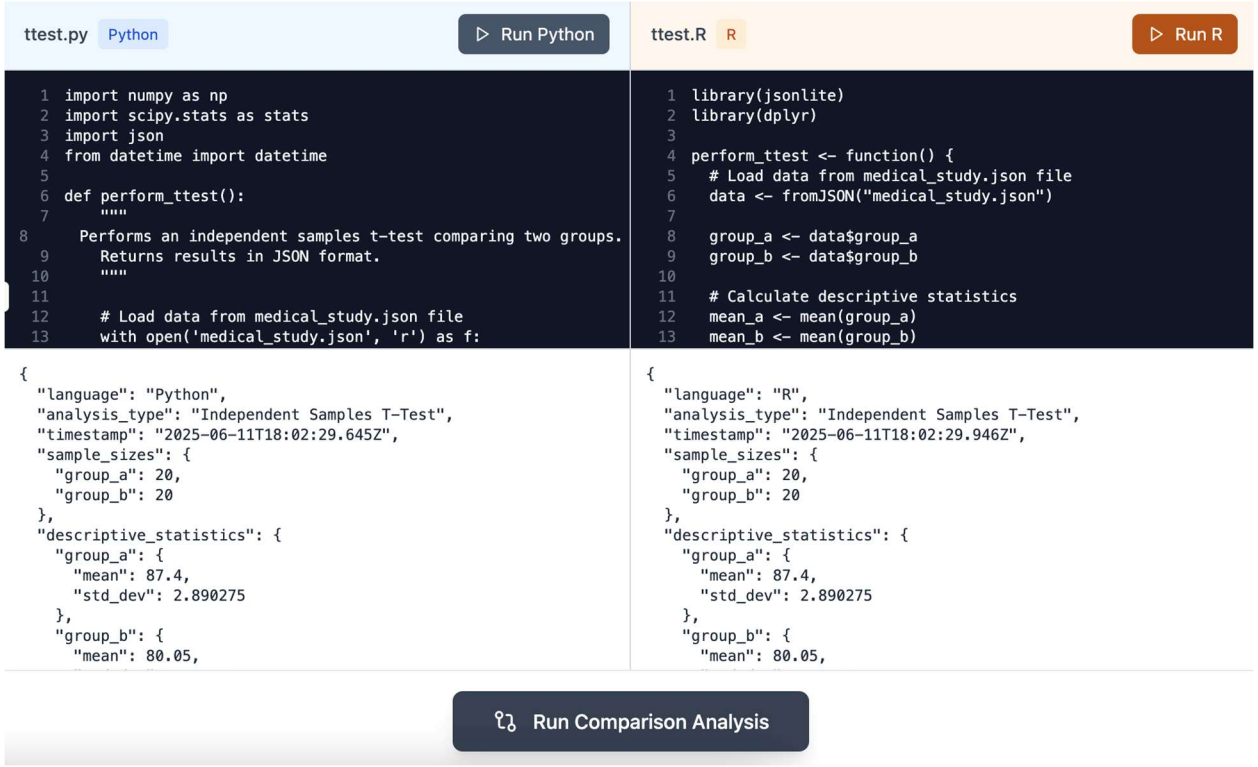
Submit


Submit File
Once submitted, file will be available for review and you may receive comments.
Study
Select study
Deliverable
Select deliverable
Stage
Select stage
Status
Select status
I followed EDA's SOP-001 QC Plan
I followed EDA's WI-042 Work Instruction
Submit the file

Seed


ttest.py
Python
Run Python
import numpy as np
import scipy.stats as stats
import json
from datetime import datetime
def perform_ttest():
"""
Performs an independent samples t-test comparing two groups.
Returns results in JSON format.
"""
Load data from medical_study.json file
with open('medical_study.json', 'r') as f:
data = json.load(f)
group_a = data['group_a']
Run Python analysis to see results
ttest.R
R
Run R
library(jsonlite)
library(dplyr)
perform_ttest <- function() {
# Load data from medical_study.json file
data <- fromJSON("medical_study.json")
group_a <- data$group_a
group_b <- data$group_b
# Calculate descriptive statistics
mean_a <- mean(group_a)
mean_b <- mean(group_b)
sd_a <- sd(group_a)
sd_b <- sd(group_b)
Run R analysis to see results
Run Comparison Analysis

Compare



Roadmap: Validation

- SOPs + user docs
- Verify test modules (t-test, ANOVA)
- Output audit trail across tools
- Regulatory-ready (SOC2, Part 11)

Integration Strategy

- Microsoft Enterprise ID
- Azure container
- API- first (integrate with other tools)
- MS Graph API for SharePoint, Teams, GitHub

Our Differentiators
Real double programming

- Append-only event log automates tracker
- Role-based views to maintain independence
- Compare engine across R, Py, SAS
- Cloud-native, Microsoft-integrated

Thank You

TP - SOP & Work Instructions
JJ – Product & Architecture
FR – Frontend + UI
XH – System + Reconciliation

Let's talk QC. cal.com/eda

Demo

- Viewer Demo
- API Demo

- **EDA for QC of ADaM**