MANAGING SAMPLE FAILURES IN CLARITY LIMS

Chris Patterson
Workflow design by Mark Swinkels

Semaphore Solutions
www.semaphoresolutions.com
1 (844) 744-3577 ext 1

MANAGING SAMPLE FAILURES IN CLARITY LIMS

INTRODUCTION

At the end of a sample or library prep protocol, a laboratory will typically want to evaluate the quality of the work in progress.

For samples that meet the Quality Control (QC) standards, the lab worker's action is generally straight-forward; move the samples on to the next protocol in the workflow.

For samples that fail QC, however, there is often complex decision-making or consultation that needs to occur. This can be a challenge to correctly model in your Clarity workflow.

This paper describes a collection of steps and scripts that, when used together, provides a process for handling of samples that fail QC, including:

- A manager-evaluation workflow path that is flexible and extensible, supports notifications, and doesn't impede the progress of samples that have passed QC.
- A pooling workflow path that can aggregate the outputs of multiple prep runs of the same sample.

Also included is a process for handling, in Clarity, failure scenarios that fall outside of the normal workflow, such as a dropped container, or spoiled sample.

THE STARTING POINT

In a typical Clarity protocol, the prep work will lead down to a single QC step at the end (Figure 1). Samples that pass QC will move on to the next protocol, and failed samples will go through Clarity's built-in manager review process, where the manager can choose from continuing the samples on, reworking them from a previous step, or removing them from the workflow.

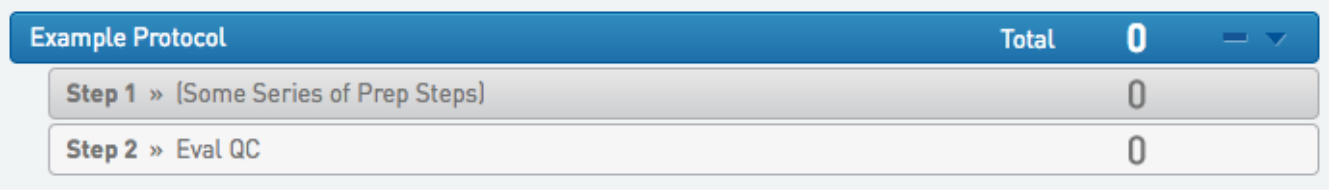

Figure 1: A Typical Protocol with QC

MANAGING SAMPLE FAILURES IN CLARITY LIMS

The built-in feature is easy to use, and works out of the box, but there are a couple of reasons why we propose a custom solution.

First, since lab managers are not typically monitoring Clarity for escalated samples to review, they generally want to be notified when there are samples that require their attention. In most other places in the Clarity workflow, the lab could add an EPP script that sends an email, but the built-in manager review feature has no step transitions to attach EPPs to. As a result, the researcher would need to notify a reviewer manually to avoid the sample sitting unnoticed.

Second, sometimes a lab manager will want to push failed samples through the workflow anyways, but also need to override UDFs that would cause issues in downstream calculations. Unfortunately, the built-in manager review screen does not permit setting or editing UDFs on the samples being reviewed.

Third, Clarity's two configuration options for handling passed samples during manager review each come with undesirable side effects. In the default configuration, passed samples move on to the next step immediately, with only failed samples held back for the manager review.

This sounds correct, but, in practice, any EPPs set up to run at the end of the step only run when the manager review has been completed. As a result, that EPP is likely running against some samples that have already moved on in the workflow, and are no longer in the same step context as the failed samples. This can have unpredictable results.

Alternately, Clarity can be configured to hold back all of the samples in a container until the Manager Review process has been finished. This avoids the problem with EPPs running on step completion, but introduces a new problem. Assuming that the reviewer has been notified, and is able to work on the failed samples, there may be times where the reviewer does not immediately know what actions are required without additional consultation. For example, a tumor may need to be examined by someone else in the lab before deciding if sample re-extraction is possible. In this case, because the entire container is being held up in review, passed samples that could in fact be worked on would sit waiting for this consultation to be completed.

MANAGING SAMPLE FAILURES IN CLARITY LIMS

ADDING A CUSTOM REVIEW STEP

In our solution we add a new, result-file-output Manager Review step in a separate protocol, and separate workflow (Figure 2). We then add an EPP to the end of Eval QC that automatically removes failed samples from whatever workflow they're currently in, and queues them instead to the new Manager Review workflow.

Additionally, the EPP sends a notification to any lab managers that might need to review the failed samples, by whatever method fits the lab's communication process. It could be an email to a person selected from a drop-down step UDF, or a message posted to a specific user in the lab's chosen chat client.

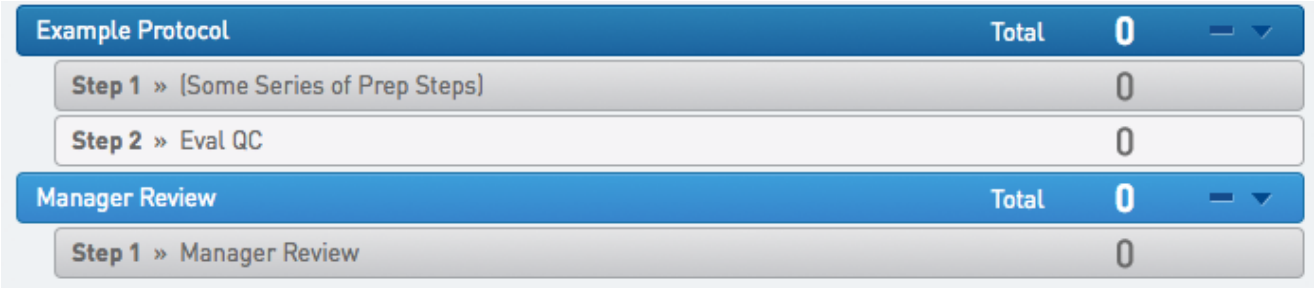

Figure 2: Manager Review in Place

In this review pattern, any additional EPPs running on the completion of Eval QC can run when all samples are together in the correct context. The lab tech can continue to work on passed samples, while failed samples wait in the queue of Manager Review for the appropriate user, who has just been notified that there are samples requiring attention.

Next, we configure the Manager Review step to handle whatever options the lab wants for failed samples.

It could be pushing a sample through regardless of its QC result, reworking an ancestor artifact back from an earlier point in the workflow, or triggering a message to a CRM system to notify a customer that their sample has failed.

While the specifics of the actions at the end of the step will be individual to each lab, the pattern involves the following steps:

MANAGING SAMPLE FAILURES IN CLARITY LIMS

- Create a result file UDF, with specified presets, to let the reviewer select a review action. Each preset represents an option for the sample's progress.
- Create an EPP running on completion of the step, mapping the review action to a piece of work, typically being one or more of:
  - Continue the sample forward.
  - Drop the sample from the workflow.
  - Rework an ancestor artifact back into an earlier part of the workflow.
  - Make an API call to another linked system.
  - Send a notification to some party outside of the lab.
- Add a UDF for the researcher to add comments about why a sample is being sent for review.
- Add a second UDF for the reviewer to add comments about the review decision.

ONE SMALL THING

The one downside to this custom review pattern is the loss of the built-in feature's user permissions control. Clarity's manager review forces you to select a user to review the failed samples, and only that user may complete the step.

To get similar functionality in the custom review step, you would need to recreate it yourself, using a step UDF on Eval QC to replace the selected reviewer, along with a custom EPP to validate the logged-in user's name, or role, when starting samples in Manager Review.

ADDING A STEP TO HOLD SAMPLES FOR POOLING

When a sample fails QC due to low volume or concentration, a lab may want to combine it with a second sample run, in order to bring up the failing level.

To set up this new workflow path, we first need a place to put samples to wait to be pooled, so add a new Pool Samples step before Eval QC (Figure 3). This step should be configured without indexing, as members of these pools will not need to be individually identified later.

Second, wire in the route from Manager Review to this new step. Add a new review action preset to the result file UDF, something like 'Hold for Pooling'. Add a matching handler to the EPP that runs on completion of Manager Review that routes the current sample to the new Pool Samples step, and reworks one of the sample's ancestor artifacts back into an appropriate step earlier in the workflow.

MANAGING SAMPLE FAILURES IN CLARITY LIMS

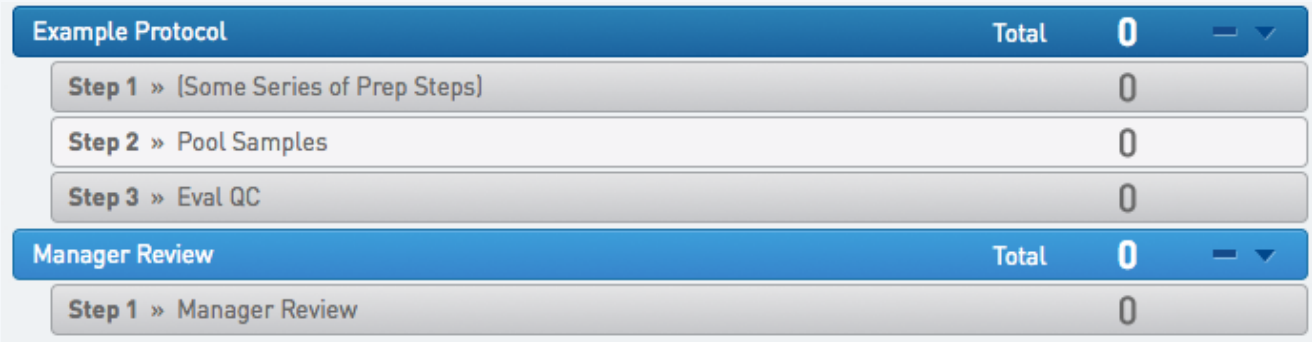

Figure 3: Protocol Steps Complete

Third, add a new EPP at the end of the final prep step that automatically routes samples to the Pool Samples step if a sibling (artifact from the same submitted sample) is found in the queue there, so they can be pooled together. Once pooled, the combined sample can then be run through Eval QC again, and it may now have sufficient volume or concentration to pass.

ALTERNATE CONFIGURATION

An alternate place to put the Pool Samples step is after individual QC has been done. If placed before QC, the lab would re-run quantification on the combined sample, but if one of the individual runs resulted in a bad sample, the combined pool may also be spoiled.

If placed after QC, the lab would be safe from mixing good samples with bad, but calculating the combined concentration would require volumetric measurements that may or not actually be captured anywhere. Either type, even both types, of pooling may be used in a lab.

The downside of placing Pool Samples at the end of the protocol is that the user experience suffers somewhat. Because Clarity does not permit ending a protocol from a step that is not the last step, normal samples leaving Eval QC would need to have their next steps set to 'Remove from workflow', with an EPP manually routing them to the first step of the next protocol. Moving post-QC Pool Samples into the Manager Review protocol after the Manager Review step would be a better solution, as samples

MANAGING SAMPLE FAILURES IN CLARITY LIMS

finishing Manager Review are already expected to be removed from that workflow, and an EPP for custom sample routing is already part of the step.

HANDLING SAMPLE FAILURES OUTSIDE THE NORM

So far we've covered how to manage samples that fail quality control during the normal flow through the Clarity workflow, but some failures happen outside of the context of a step, at any point in the workflow. A lab tech may drop a container, or leave it outside of refrigeration to spoil. A sample on a plate may be accidentally contaminated.

To cover these scenarios, we can leverage a lesser-known Clarity function, which is the Derived Sample Automation (DSA). In simple terms, a DSA is a user-triggered EPP script that runs against artifacts anywhere in the workflow, as opposed to a regular EPP which runs only in the context of an in-progress step.

A simple 'remove sample for review' DSA would cover all of our scenarios, but a second 'remove container for review' DSA would improve the user experience for container-level failure by eliminating the need to carefully multi-select a large number of samples in the UI.

When selected to run against a queued artifact, the scripts would behave as follows:

- Remove Sample for Review – The UI prompts the user with a 'Reason for Review' text field. The script removes the sample from the step to which it's currently queued, and routes the sample to the queue of the Manager Review step. The 'Reason for Review' is recorded as a comment UDF.
- Remove Container for Review – Like the previous DSA, except that all samples in the same container that are queued to the step are affected.

The underlying EPP for both DSAs can be the same for both cases, with a script flag signaling whether the removal is container-wide or not.