Master of Technology

Individual Project Report

DRAM Customer Return System

by

Ong Boon Ping

**1. Abstract**

In a particular DRAM manufacturing factory, QA is in charge of receiving customer returned parts. While QA is able to re-run the testing flow, it is relied on test engineering department to provide further technical intent of the failing parts.

After gathering the info, QA will decide further industrial action to be taken based on the finding of customer returned part.

**2. Problem Description**

QA lacks of a multi-user platform that allows QA, test engineering and sales representatives to key in each findings. The platform should be able to run with simple rules that can help QA to do decision making.

This platform should save time of coordinating works and decision making.

**3. Knowledge Modelling**

**3.1 Knowledge identification**

The knowledge is identified from QA customer return procedure and interviews.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Source | Insights from the Source | Acquisition Technique |
| 1 | QA | How returned part cases can be closed:   1. Degrader 2. Guardbanded Parts 3. Temperature Violation 4. Additional testing 5. Customer system analysis 6. Requesting parts with higher specification | Manual extraction from Customer Return Procedure |
| 2 | DRAM Test Engineering | Type of analysis that can be done on the returned parts | Interviewing test engineers |
| 3 | DRAM Sales | Customer communication ways based on the analysis result. | Interviewing sales personnels |

**3.2 Knowledge Specification**

Based on closed cases in the past, we will categorize cases based on various testing/analysis that can be done

1. Rerun=Fail AND PastResult Exists AND !FlowChanged -> Degrader
2. Rerun=Fail AND PastResult Exists AND FlowChanged -> Guardbanded Parts
3. (Rerun=Pass || PastResult NOT EXISTS) && ExtendedTemperatureFlow ==Fail -> Temperature Violation
4. (Rerun=Pass || PastResult NOT EXISTS) && ExtendedTemperatureFlow ==Pass && StringentTesting==Fail && ProductionOverkillPercentage<1.0 -> Additional testing
5. (Rerun=Pass || PastResult NOT EXISTS) && ExtendedTemperatureFlow ==Fail && StringentTesting==Pass -> Customer system analysis
6. (Rerun=Pass || PastResult NOT EXISTS) && ExtendedTemperatureFlow ==Pass && StringentTesting==Fail && ProductionOverkillPercentage>1.0 -> Requesting parts with higher specification

**4. Workflow and Features**

The rules are the basis of the KIE workflow. With the activity diagram, the process is being defined. Ruleflow group is then assigned to the rule task respectively. User QA, TestEngineer and DRAMSales are defined.

**4.1 Activity Diagram**

DRAMSales

System

TestEngineer

QA

START

Part Detail Requesting parts with higher specification

Rerun Result

Past History and Flow Changes

Guardbanded/Degraders?

Yes

No

Close as Degrader or Guardbanded Parts

Extended Temperature Flow Result

1

END

1

Failing temparature flow?

Yes

Close as Temperature Violation

No

Stringent Flow Result

Passing?

No

Yes

END

Check with customer on system

Overkill Result on limited samples

Overkill < 1.0%?

Yes

END

Close with additional test

No

Check with customer on higher specifications

END

END

**4.2 Data Object**

|  |  |  |
| --- | --- | --- |
| Data Object | Field | Type |
| returnPartDetail | serialID | String |
| returnPartDetail | customerName | String |
| flowRerunResult | flowRerunPassFail | Boolean |
| flowChange | flowChanged | Boolean |
| pastPartHistory | pastPartPassFail | Boolean |
| extendedTemperature | extendedTemperaturePassFail | Boolean |
| stringentFlow | stringentFlowPassFail | Boolean |
| productionOverkill | productionOverkillPercent | Float |

**4.3 Features**

The system is designed for the QA to record the return part details and it will trigger test engineers to check on the parts.

At the beginning, QA will key in the part detail and the test re-run (on the part) result.

Test Engineer will check whether the past data is available and the part is passing.

If the part is passing re-run, the analysis will be continued by test engineers. If it fails re-run and past testing record still in test engineering database, if the testing flow remain unchanged, the part will be considered as degrader. However, if it fails re-run while the testing flow has been changed after the parts goes to customer, it will be defined as guardbanded parts. QA will get the result and enter the comment for degrader/guardbanded parts.

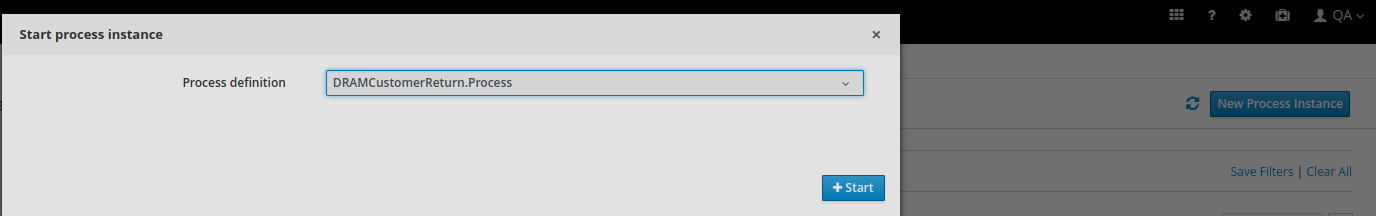
However, if the parts passing the re-run, analysis continues. In common, customer system can get too hot and so the parts fail. Thus, temprature range is extended for the test flow. If it fails, the parts is ceonsidered to be a customer violation case. If it passes, the returned part will be going for further analysis with stringent test flow. If it pass the stringent flow, sales team should discuss with customer on whether the customer system has limitation or abusing the parts.

If it fails stringent test, the test engineer will run the stringent test on limited samples. If the stringent test will results in fallout (>1%), the sales representative should discuss with customer in getting higher performance parts and specify new requirement. If the overkill fallout is acceptable, QA can close the case while TE can continue to implement the test flow.

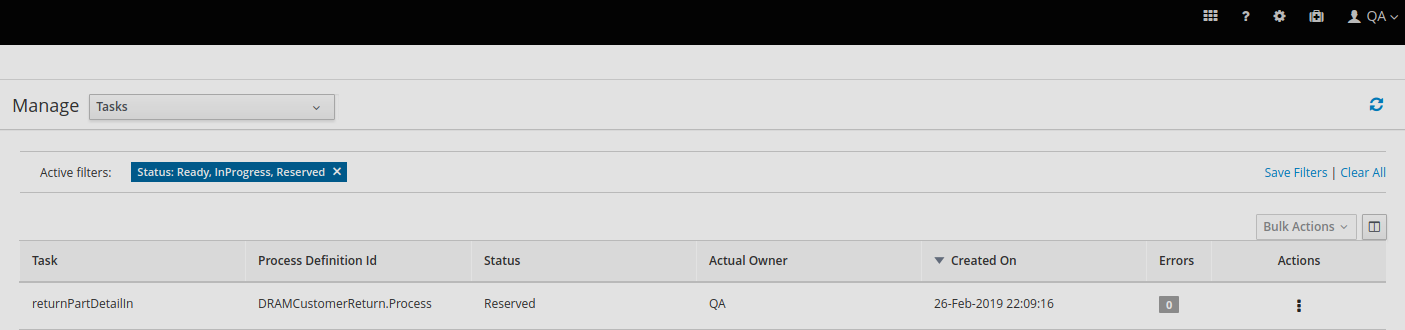
**5. Validation**

Validation Snapshots:

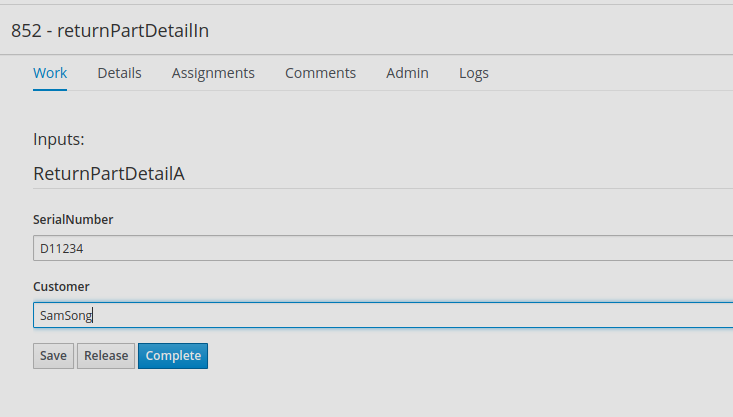
Start by user “QA”

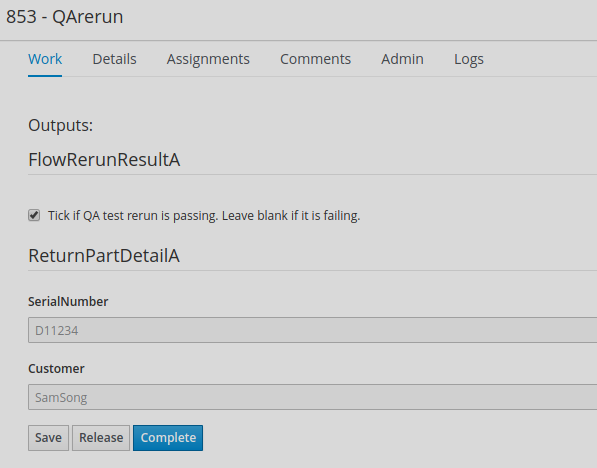


QA receives task

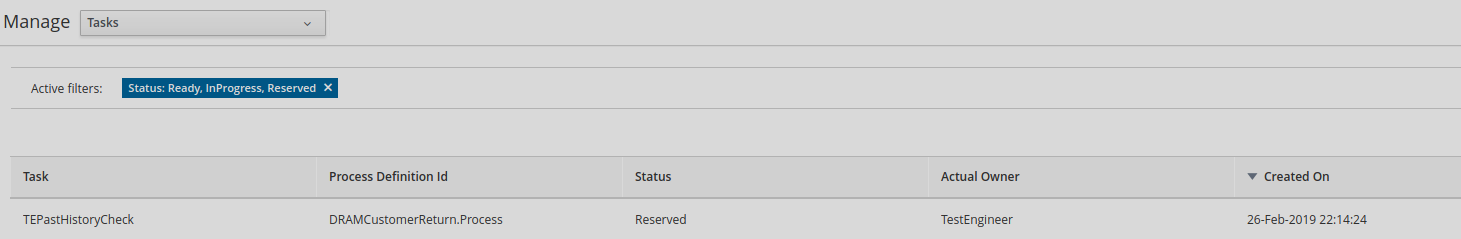


QA fills in part detail and flow rerun result

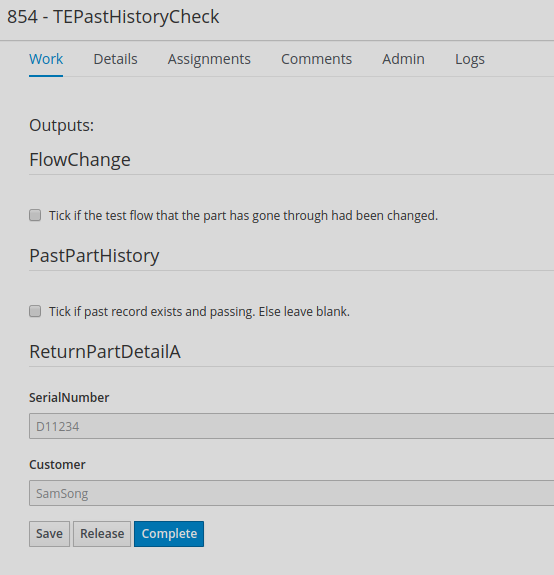


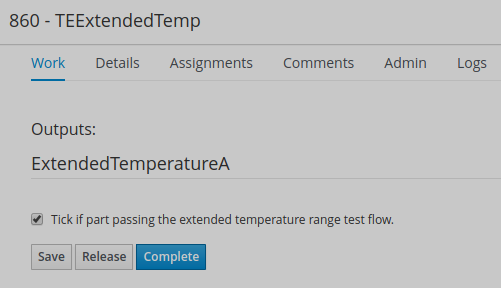


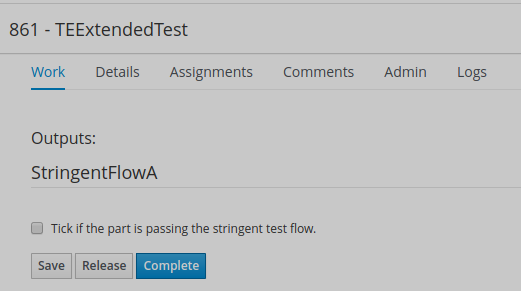
TestEngineer gets the task

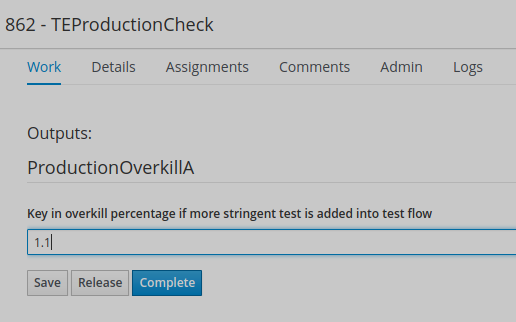


TestEngineer fills in past part history, extended temperature flow (after proved that the part is not resolved by re-run or past history given), stringent flow and production overkill result.

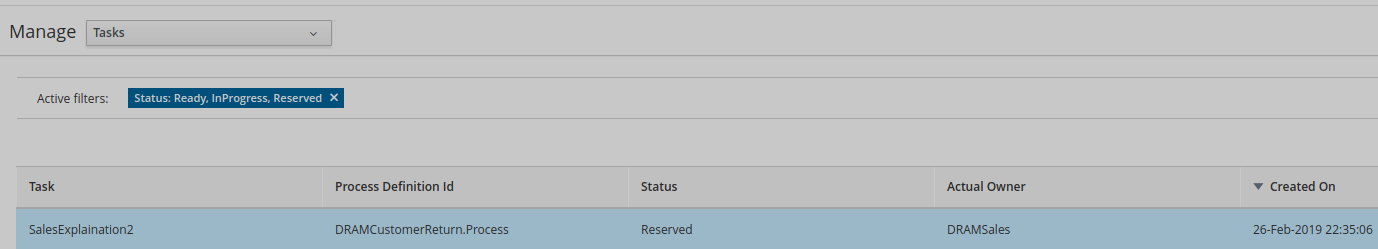


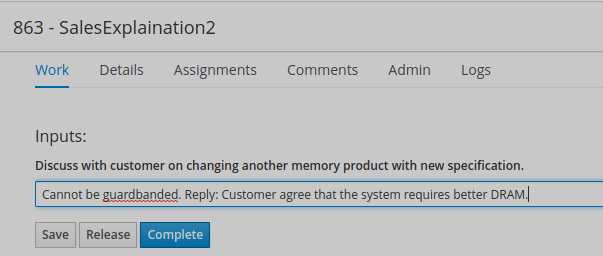






DRAMSales get the task when the additional test is having high overkill.





**6. Conclusion**

The system was validated and it is able to run with the expectation. (Using KIE 7.16 downloaded from JBOSS, space name individual and project DRAMCustomerReturn).